

# WIND User's Guide\*

The NPARC Alliance

*NASA Glenn Research Center*

*Cleveland, Ohio*

*USAF Arnold Engineering Development Center*

*Tullahoma, Tennessee*

---

\*This document describes the use of WIND Version 4.0, and was created November 15, 2002. It was originally based on the 24 Jan 1997 version of the *NASTD User's Guide*, by R. H. Bush, M. Mani, T. R. Michal, and W. W. Romer of McDonnell Douglas Aerospace. Please send documentation suggestions/corrections to Charlie Towne at [towne@grc.nasa.gov](mailto:towne@grc.nasa.gov). Questions about the WIND code itself or the NPARC Alliance should be sent to [nparc-support@info.arnold.af.mil](mailto:nparc-support@info.arnold.af.mil), (931) 454-7455



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Mathematical Model	1
1.2	Geometry and Mesh Description	1
1.3	Numerical Technique	1
1.4	Coding	2
<b>2</b>	<b>Tutorial</b>	<b>3</b>
2.1	Gather Information	3
2.1.1	Geometry	3
2.1.2	Flow Conditions	4
2.1.3	Boundary Conditions	4
2.2	Create the Computational Grid	5
2.2.1	General Requirements	5
2.2.2	Creating the Common Grid (.cgd) File	6
2.2.3	Test Case 4 Grid	6
2.3	Define the Input	11
2.3.1	Descriptive Header and Comments	11
2.3.2	Boundary Conditions	12
2.3.3	Flow and Initial Conditions	12
2.3.4	Physical Model Controls	12
2.3.5	Numerical Model Controls	13
2.3.6	Test Case 4 Input	15
2.4	Run the Code	18
2.4.1	The <i>wind</i> Script	18
2.4.2	Parallel Operation	19
2.4.3	Running Test Case 4	19
2.5	Monitor Convergence	21
2.5.1	Test Case 4 Convergence	22
2.6	Examine the Results	25
2.6.1	Using CFPOST	25
2.6.2	Test Case 4 Results	25
2.7	Summary	26
<b>3</b>	<b>Geometry and Flow Physics Modeling</b>	<b>29</b>
3.1	Symmetry Considerations	29
3.1.1	Three-Dimensional Cases	29
3.1.2	Two-Dimensional Cases	29
3.1.3	Area Variation (Quasi-3D) Cases	29
3.1.4	Axisymmetric Cases	29
3.2	Euler and Navier-Stokes Equations	30
3.2.1	Freestream Conditions	30
3.2.2	Reynolds Number Considerations	30
3.2.3	Mass Flow in Two-Dimensional Calculations	32
3.2.4	Mass Flow and Grid Areas	33
3.2.5	Heat Transfer	33
3.2.6	Viscosity	33
3.3	Turbulence Models	33
3.3.1	Algebraic Models	34
3.3.2	One-Equation Models	34

3.3.3	Two-Equation Models	34
3.3.4	Transition Specification	35
3.4	Gas Models	35
3.5	Flowfield Initialization	35
3.5.1	User-Specified Initialization	35
3.5.2	Boundary Layer Initialization	36
3.5.3	Reinitialization	36
<b>4</b>	<b>Numerical Modeling</b>	<b>37</b>
4.1	Iterations and Cycles	37
4.1.1	“Turning Off” Zones	37
4.2	Grid Considerations	37
4.2.1	Grid Sequencing	37
4.2.2	Thin-Shear-Layer Calculations	38
4.3	Explicit Operator	38
4.3.1	Explicit Smoothing	38
4.3.2	Explicit Boundary Damping	39
4.3.3	Total-Variation-Diminishing (TVD) Operator	39
4.4	Implicit Operator	39
4.4.1	Implicit Boundaries	39
4.5	Time Step	40
4.5.1	CFL Number	40
4.5.2	Runge-Kutta Time Step	41
4.5.3	Global Newton Iteration	41
4.5.4	Time-Accurate Solutions	41
4.6	Convergence Acceleration	42
<b>5</b>	<b>Boundary Conditions</b>	<b>43</b>
5.1	Explicit and Implicit Boundary Conditions	43
5.2	Boundary Condition Types	43
5.3	Wall Boundary Conditions	44
5.3.1	Inviscid Wall	44
5.3.2	Viscous Wall	44
5.3.3	Bleed	45
5.4	Flow Interface Boundary Conditions	45
5.4.1	Freestream	45
5.4.2	Arbitrary Inflow	46
5.4.3	Outflow	46
5.5	Grid Topology Boundary Conditions	47
5.5.1	Reflection	47
5.5.2	Self-Closing	47
5.5.3	Singular Axis	47
5.5.4	Pinwheel	48
5.6	Zonal Interface Boundary Conditions	48
5.6.1	Coupled	48
5.6.2	Chimera	50
5.7	Miscellaneous Boundary Conditions	50
5.7.1	Undefined	50
5.7.2	Frozen	50
<b>6</b>	<b>Convergence Monitoring</b>	<b>51</b>

6.1	Residuals	51
6.2	Integrated Flowfield Quantities	51
6.3	History Tracking of Flow Data	52
<b>7</b>	<b>Files</b>	<b>53</b>
7.1	Input Data File (.dat)	53
7.2	Grid File (.cgd)	53
7.3	Flow File (.cfl)	54
7.4	Global Newton File (.cfk)	54
7.5	Boundary Data File (.tda)	54
7.6	Time History File (.cth)	55
7.7	List Output File (.lis)	55
7.8	WIND Stop File (NDSTOP)	55
7.9	Temperature and Transition Specification Files	56
7.10	Chemistry Files (.chm)	56
7.10.1	Header	56
7.10.2	Thermodynamic Coefficients	57
7.10.3	Finite Rate Coefficients	58
7.10.4	Transport Coefficients	61
7.11	Reserved Files	62
<b>8</b>	<b>Scripts</b>	<b>63</b>
8.1	<i>wind</i> — Run WIND Code	63
8.2	<i>wind_post</i> — Perform Post-Processing	66
8.3	<i>windmp</i> — Run on Multi-Processor	68
8.4	<i>windver</i> — Get WIND Version Number	68
<b>9</b>	<b>Parallel Processing</b>	<b>69</b>
9.1	Multi-Processing Control File	69
9.2	Multi-Processing Script Considerations	72
9.3	Multiple Parallel Jobs	73
9.4	Hints	73
<b>10</b>	<b>Keyword Reference</b>	<b>75</b>
10.1	Text Conventions	75
10.2	Zone Selection	75
10.3	Keyword Details	76
	ACCELERATE — Convergence acceleration (block)	77
	ACTUATOR   SCREEN — Discontinuous change across a zone boundary (block)	78
	ARBITRARY INFLOW — Arbitrary inflow (block)	83
	AXISYMMETRIC   AXI-SYM — Axisymmetric flow	91
	BLEED — Bleed region flow rate	92
	BLOW — Inject vectored flow over a selected region	95
	BL_INIT — Boundary layer initialization	96
	BOUNDARY-DAMP   BDAMP — Boundary damping (block)	97
	BOUNDARY TVD — Boundary total variation diminishing operator flag	98
	CFL# — CFL/time step specification	99
	CGNSBASE — Use CGNS files	100
	CHEMISTRY — Chemistry model selection (block)	101
	/ — Comment lines	104
	COMPRESSOR FACE — Outflow boundaries, compressor face	105

CONVERGE — Controls convergence . . . . .	108
COUPLING — Zone coupling mode specification . . . . .	109
CYCLES — Number of solution cycles . . . . .	110
CROSSFLOW — Crossflow CFL factor . . . . .	111
DOWNSTREAM MACH — Outflow boundaries, Mach number . . . . .	112
DOWNSTREAM PRESSURE — Outflow boundaries, pressure . . . . .	113
DQ — $\Delta Q$ limiter . . . . .	115
END — Termination . . . . .	116
FIXER — Instability smoothing . . . . .	117
FREESTREAM — Freestream conditions . . . . .	118
FRINGE — Solution mode at fringe points . . . . .	119
GAS — Gas property specification . . . . .	120
GRAVITY — Add gravity body forces . . . . .	121
GRID LIMITER — Grid limiting capability . . . . .	122
HISTORY — Time history flowfield variable tracking . . . . .	123
IMPLICIT — Implicit operator control . . . . .	126
IMPLICIT BOUNDARY — Implicit boundary conditions . . . . .	128
INCLUDE — Include a file in the standard input . . . . .	129
ITERATIONS — Set number of iterations per cycle . . . . .	130
LOADS — Flowfield integration (block) . . . . .	131
MARCHING — Parabolized Navier-Stokes algorithm . . . . .	134
MASS FLOW — Outflow boundaries, mass flow . . . . .	135
MFD — Magneto-Fluid Dynamics Model (block) . . . . .	136
MOVING WALL — Specify moving wall boundaries . . . . .	141
NAVIER-STOKES ITERATIONS — Navier-Stokes sub-iterations . . . . .	142
NEWTON — Use Global Newton time stepping . . . . .	143
OUTFLOW NON-REFLECTING — Outflow boundaries, non-reflecting . . . . .	144
PERIODIC — Periodic boundaries . . . . .	145
REINITIALIZE — Reinitialize selected flowfield zones on restart . . . . .	146
RELAX COUPLING — Set zone coupling relaxation factor . . . . .	147
REL-ROT-ZONE — Relative rotating zones (block) . . . . .	148
RESTART   START — Begin run in specified zone . . . . .	153
RHS — Explicit operator control . . . . .	154
ROLL — Specify roll about one of the coordinate axes . . . . .	155
ROTATE — Perform calculation in a rotating frame of reference . . . . .	156
SEQUENCE — Grid sequencing control . . . . .	157
SMOOTHING — Add dissipation to explicit operator . . . . .	158
SPAWN — Run an external process from WIND . . . . .	159
STAGES — Multi-stage time stepping . . . . .	161
TEST — Non-production test options . . . . .	162
TSL   THIN SHEAR LAYER — Thin shear layer option . . . . .	163
TTSPEC — Wall temperature and transition (block) . . . . .	164
TURBULENCE — Turbulence model selection . . . . .	167
TVD — Total Variation Diminishing operator flag . . . . .	176
VISCOSITY — Specification of viscosity law . . . . .	177
WALL FUNCTION — Specify the use of wall functions . . . . .	178
WALL SLIP — Iterations until no slip . . . . .	179
WALL TEMPERATURE — Specify wall temperature . . . . .	180

References	195
------------	-----





# 1 Introduction

This manual describes the operation and use of the WIND code, a computational platform which may be used to numerically solve various sets of equations governing physical phenomena. WIND represents a merger of the capabilities of three older CFD codes — NASTD (the primary flow solver at McDonnell Douglas, now part of Boeing), NPARC (the original NPARC Alliance flow solver), and NXAIR (an AEDC code used primarily for store separation problems).<sup>1</sup> Currently, the code supports the solution of the Euler and Navier-Stokes equations of fluid mechanics, along with supporting equation sets governing turbulent and chemically reacting flows.

WIND uses multi-zone computational grids, and is capable of computing solutions on a wide variety of grids. Because WIND is written to accommodate arbitrary grid topologies and boundary condition combinations, it may be used to obtain solutions about most of the geometric configurations for which a grid can be generated.

The multi-zone approach to flow solutions makes it possible to decompose virtually any configuration into a number of manageable subregions, or zones. Zonal connectivity information is computed using a pre-processing Grid MANipulation (GMAN) code, and stored in the grid file used by WIND. During the course of a solution, WIND maintains continuity in flow properties across zone boundaries through a process known as zone coupling (Romer and Bush, 1993).

## 1.1 Mathematical Model

All terms are retained in the governing equations, including secondary flow, reversed flow convection, pressure gradients normal to a wall, streamwise diffusion, and unsteady flow. All heat transfer terms are retained. Several algebraic, one-equation, and two-equation turbulence models are available. Transition may be specified through the use of an external file. Modification of the effective heat transport coefficient due to turbulence is linked to the momentum diffusion coefficient by a turbulent Prandtl number, which is taken to be constant.

The fluid may be treated as a thermally and calorically perfect gas, a thermally perfect gas, equilibrium air, or a mixture undergoing a finite rate chemical reaction. For ideal gas, conventional values are given to the gas constant ( $R$ ) and the ratio of specific heats ( $\gamma$ ), or they may be specified. Effects of gravity on the fluid (i.e., stratification) are not included.

## 1.2 Geometry and Mesh Description

WIND uses externally generated computational grids. Therefore, all geometric input and capability depend on the grid generator. WIND has no geometric input. All analyses must be preceded by a grid generation run.

## 1.3 Numerical Technique

The solution is executed iteratively on the computational mesh. The flow equations are evaluated using second-order-accurate finite differences. The partial differential equations are modeled in their conservative form. Explicit terms are computed using either upwind or central differencing, and

---

<sup>1</sup>WIND is a product of the [NPARC Alliance](#), a partnership between the [NASA Glenn Research Center \(GRC\)](#) and the [USAF Arnold Engineering Development Center \(AEDC\)](#) dedicated to the establishment of a national, applications-oriented flow simulation capability. [The Boeing Company](#) has also been closely associated with the Alliance since its inception, and represents the interests of the NPARC User's Association.

their order may be controlled through the use of keywords in the input data file. The implicit terms are computed using either an approximately factored or four-stage Runge-Kutta scheme, or they may be disabled altogether. A Global Newton iteration scheme is also available, and may be used for unsteady flows with large time scales or as a convergence acceleration technique for steady flows.

### 1.4 Coding

WIND is coded in the Fortran 77, Fortran 90, and C programming languages. The production version of the code is supported on a variety of systems from Silicon Graphics, Hewlett-Packard, Sun, and Cray.

## 2 Tutorial

This section is intended primarily for new users to demonstrate the simulation process using WIND. More experienced users may find this section useful as a road map through the simulation process and to help demonstrate new features. The approaches presented here are by no means unique, and detailed information is excluded by design. The user is referred to later sections of this User's Guide for more detailed information on the various aspects of running the WIND code, and in particular to [Section 10](#) for more in-depth discussions on the choices available for each input keyword.

The approach taken here is to discuss the flow simulation process using as an example a simple subsonic internal flow in a diverging duct. The various files discussed in this section may be downloaded from the WIND documentation WWW site, at <http://www.grc.nasa.gov/www/winddocs/user/tutorial.html#tutorial:downloading>.

While it is clearly impossible to demonstrate every option in WIND with a single application, the basic mechanics of using the WIND code are demonstrated with this case. Additional abbreviated examples are also provided in [Section 10](#) for specific keywords. For details of the flow simulation process for more complex cases, please see the various example applications accessible from the WIND Validation home page at <http://www.grc.nasa.gov/www/wind/valid/validation.html>.

The solution process using any conventional time-marching Navier-Stokes code is basically the same, and may be divided into the following steps:

- [Gather information](#)
- [Create the computational grid](#)
- [Define the input](#)
- [Run the code](#)
- [Monitor convergence](#)
- [Examine the results](#)

The mechanics of doing each of these steps may vary from code to code, however. The following sections describe how these steps are typically accomplished when using the WIND code.

### 2.1 Gather Information

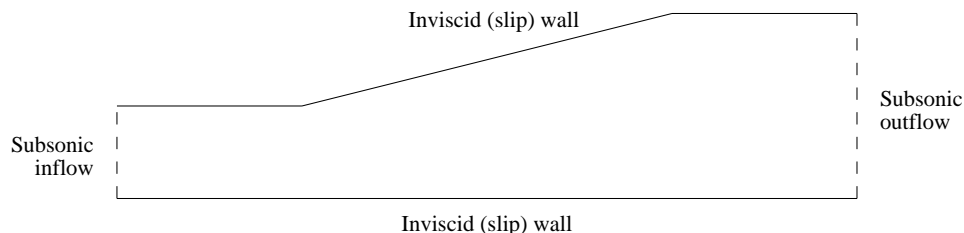
As for any project, the first step is to gather all of the information required to completely specify the problem to be analyzed. Of course, as the flow simulation process proceeds, missing information will become apparent. The required information can be divided into three major categories — [geometry](#), [flow conditions](#), and [boundary conditions](#). It is also important to understand the ultimate goal of the simulation. For example, is an accurate drag prediction required? Or, is lift required, but only to within 5%? The answers to these types of questions will determine the detail of the input information required to provide the necessary level of detail and accuracy in the solution.

#### 2.1.1 Geometry

The more geometric details that can be determined for the target application, the more likely the results will provide an accurate simulation of the flow field. This is not to say that all geometric

components must be modeled. Resolving fine geometric details of a configuration requires more grid points, and, as a result, longer run times. The level of detail to which the geometry must be modeled depends on the type of results required and the acceptable turn-around time.

The geometry of the Test Case 4 example is shown in [Figure 1](#). The duct is 8 inches long, and the entrance and exit heights are 1 inch and 2 inches, respectively.



**Figure 1:** Test case geometry

The desired result from this calculation is the pressure distribution along the upper and lower walls, and the mass flow rate to within 10%. Thus, resolving the sharp surface gradients at the corners is not necessary. Other detailed geometric features such as weld joints, for example, are also not modeled. If the fine details of the boundary layer in the vicinity of the joints were important, then a significantly more detailed geometry description would be required.

### 2.1.2 Flow Conditions

In addition to the geometric information, flow conditions are also required, and are used to set the reference conditions used in non-dimensionalizing the governing equations solved by the WIND code. As with the geometric information, the simulation results will only be as good as the flow condition information provided by the user. Flow conditions should be specified that are representative of the flow being solved, so that the nondimensional variables used in the code are on the order of 1.0. A good choice is the inlet conditions for internal flows, and the freestream conditions for external flows.

For this example case, the inlet Mach number, total pressure, and total temperature are 0.78, 15 psi, and 600 °R, respectively. Starting from these conditions, the Reynolds number based on the duct height may be computed as  $3.023 \times 10^5$ .

### 2.1.3 Boundary Conditions

Information is also required at boundaries that are at the “outer edges” of the computational domain. These boundary conditions are used to model the interaction between the flow inside the computational domain and surfaces or flows outside the domain. In fact, the boundary conditions are perhaps the most important factor influencing the accuracy of the flow computation.

Conditions at flow interface boundaries — boundaries between flows inside and outside the computational domain such as inflow, outflow, and freestream boundaries — must be known to the level of accuracy required by the simulation. For example, if flow rates are required to within 0.1%, even slight variations in total pressure at the inflow boundary must be specified. The number of conditions to be specified at a flow interface boundary depends on whether the flow is entering or leaving the computational domain, and whether it is subsonic or supersonic.

Information must also be specified at surface interface boundaries, such as solid walls and bleed regions. Simply specifying the type of boundary, such as an adiabatic no-slip wall, is often sufficient. Additional information may also be required, though, such as the wall temperature. The level of detail that is needed for this information is determined, as discussed above, by the level of detail and accuracy required in the results.

Note that other types of boundaries may be present within the overall computational domain, that are not at the “outer edges.” Multi-zone problems will have zonal interface boundaries. Some configurations will also have boundaries resulting from the grid topology, such as self-closing and singular axis boundaries. These types of boundaries need only be labeled.

For Test Case 4 the flow at both the inflow and outflow planes will be subsonic. Three conditions are needed at the inflow boundary, and one is needed at the outflow boundary. At the inflow boundary, uniform flow is specified, with total pressure and temperature equal to the inlet values of 15 psi and 600 °R. Since extreme accuracy in the solution is not needed, constant total conditions at the inflow are sufficient. At the outflow boundary, the exit static pressure is set to 14.13 psi. The Reynolds number for Test Case 4 is large enough that the boundary layers will have little influence on the pressure distribution within the duct. The upper and lower boundaries are therefore specified as inviscid (slip) walls.

In this tutorial, the procedure used to set boundary conditions when running WIND is discussed in [Section 2.3.2](#). Additional details on all the boundary conditions available in the WIND code are presented in [Section 5](#).

## 2.2 Create the Computational Grid

WIND uses externally-generated structured grids. The grids for all the zones must therefore be created before running the WIND code. The geometry of the application governs the overall shape of the boundaries, but the approach to gridding the flow field is not unique.

### 2.2.1 General Requirements

The WIND flow simulator provides considerable flexibility. Grid lines can conform to complex shapes or may pass through regions not in the flow field. The grid may be divided into zones to conform to the geometry better, to allow grid embedding (i.e., zones with finer grids in regions of high gradients like boundary layers), and/or to allow parallel computation. These zones may be abutting or overlapping, and overlapping grids may be single- or double-fringed.

WIND requires structured grids, in which the indices  $(i, j, k)$  represent a curvilinear coordinate system, and physical Cartesian coordinates  $(x, y, z)$  are defined for each integer combination of indices. The handedness of both the physical and curvilinear coordinate systems is required to be the same at all points in the grid, i.e., both must be either left-handed or right-handed. Additionally, at least three grid points must fall between any two grid lines which represent a boundary within the computational domain. For example, if the  $k = 1$  boundary represents a solid surface and an adjacent  $k$  boundary represents a symmetry plane, the symmetry plane must be at  $k = 5$  or higher, so that the three points  $k = 2, 3$ , and  $4$  (at least) lie between the two boundaries.

The method used to create the grid is completely up to the user. For complex geometries, a sophisticated grid generation program is normally used. For very simple geometries, it may be easier to write a short program that constructs a grid using algebraic techniques.

## 2.2.2 Creating the Common Grid (.cgd) File

The computational grid used by WIND for a particular case is stored in a *Common Grid* (.cgd) file, so named because the file is formatted according to Boeing's Common File Format.<sup>2</sup> In this file are stored the  $(x, y, z)$  coordinates of all the grid points, zone coupling information, and grid units and scaling data.<sup>3</sup>

Since some grid generation codes do not produce .cgd files directly, a separate utility called *cfcnvrt* is included with WIND that may be used to convert a variety of file formats, including PLOT3D files, to Common Files. A typical procedure is thus to first store the grid file as a PLOT3D xyz file, which is an available option in most general-purpose grid generation codes, and then convert it to a .cgd file using *cfcnvrt*.

Zone coupling information is then added to the .cgd file using a pre-processing utility called *GMAN*. This is typically done at the same time as the boundary condition types are defined, and is discussed in Section 2.3.2. GMAN may then be used to examine the .cgd file, assessing grid quality and listing information about the points and zones in the grid. GMAN may also be used to generate the flowfield (i.e., interior) grid itself, given the grids on the zonal boundaries. The *GMAN User's Guide* contains descriptions of these capabilities, and others.

## 2.2.3 Test Case 4 Grid

The computational grid for Test Case 4 was constructed by simple algebraic techniques using the following program, called *case4.mesh.f*, which creates a PLOT3D xyz file in 2-D unformatted multi-zone form to Fortran unit 2, without an iblank array.

```

      Program mesh
      c
      c-----Purpose:  This subroutine computes a 2-D three-zone grid for a
      c                   simple diffuser, one of the test cases supplied with the
      c                   WIND computer code.  The grid is written to a file in
      c                   PLOT3D 2-D unformatted multi-zone form, without
      c                   iblank'ing.
      c
      c-----Called by:
      c
      c-----Calls:
      c
      c      Implicit none
      c-----Parameter statements
      c      Integer IDIM,JDIM      ! Max dimensions
      c      Integer NBLKS          ! Number of blocks
      c      Parameter (IDIM = 33, JDIM = 11)
      c      Parameter (NBLKS = 3)
      c-----Local variables:
      c      Integer i,j            ! Indices in x and y directions
      c      Integer iblk          ! Current block number
      c      Integer imax(NBLKS),jmax(NBLKS) ! Block grid sizes
      c      Real dx(NBLKS),dy(NBLKS)      ! Non-dim grid increments in blks

```

<sup>2</sup> WIND also supports the use of CGNS files for the grid and flow solution, using the [CGNSBASE](#) keyword. This tutorial, however, uses common files.

<sup>3</sup> See the *Common File User's Guide* for details about the internal structure of common files.

```

Real x(IDIM,JDIM),y(IDIM,JDIM),z(IDIM,JDIM)    ! Grid coordinates
Real xdiff1,xdiff2    ! x at start/end of diffuser section
Real xloc             ! Local x coordinate
Real xmax             ! x at end of duct
Real xstrt(NBLKS),xend(NBLKS)    ! Non-dim x limits of blocks
Real ymax             ! Max y at x = xloc
Real ymax1,ymax2      ! Max y at start/end of diffuser section
Real yslope           ! Slope of diffuser upper wall
Real ystrt(NBLKS),yend(NBLKS)    ! Non-dim y limits of blocks

c
c-----Define geometric parameters
c
      Data xdiff1,xdiff2,xmax /2.0, 6.0, 8.0/
      Data ymax1,ymax2 /1.0, 2.0/

c
c-----Set relative sizes and grid increments for each block
c
c-----Block 1
      imax(1) = 17
      jmax(1) = 6
      xstrt(1) = 0.0
      ystrt(1) = 0.0
      xend (1) = 0.5
      yend (1) = 0.5
      dx(1) = (xend(1) - xstrt(1))/(imax(1) - 1)
      dy(1) = (yend(1) - ystrt(1))/(jmax(1) - 1)
c-----Block 2
      imax(2) = 33
      jmax(2) = 11
      xstrt(2) = 0.0
      ystrt(2) = yend(1)
      xend (2) = xend(1)
      yend (2) = 1.0
      dx(2) = (xend(2) - xstrt(2))/(imax(2) - 1)
      dy(2) = (yend(2) - ystrt(2))/(jmax(2) - 1)
c-----Block 3
      imax(3) = 17
      jmax(3) = 11
      xstrt(3) = xend(1)
      ystrt(3) = 0.0
      xend (3) = 1.0
      yend (3) = 1.0
      dx(3) = (xend(3) - xstrt(3))/(imax(3) - 1)
      dy(3) = (yend(3) - ystrt(3))/(jmax(3) - 1)

c
c-----Open grid file, write header info
c
      Open (unit=2, file='fort.2', form='unformatted')
      Write (2) NBLKS
      Write (2) (imax(iblk),jmax(iblk),iblk=1,NBLKS)

c

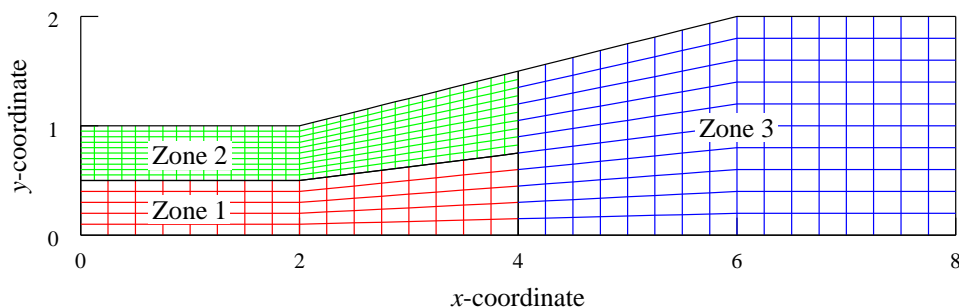
```

```

c-----Construct the grid
c
      yslope = (ymax2 - ymax1)/(xdiff2 - xdiff1)  ! Slope of diff wall
      Do 100 iblk = 1,NBLKS          ! Loop over no. of blocks
        Do 50 i = 1,imax(iblk)      ! Loop over no. of x points
c-----Compute x coordinates
          xloc = (xstrt(iblk) + dx(iblk)*(i-1))*xmax
          Do 10 j = 1,jmax(iblk)    ! Loop over no. of y points
            x(i,j) = xloc
10          Continue
c-----Get local max y, then compute y coordinates
          If (xloc .le. xdiff1) then      ! Upstream of diffuser
            ymax = ymax1
          Else if (xloc .ge. xdiff2) then  ! Downstream of diffuser
            ymax = ymax2
          Else                            ! In diffuser
            ymax = ymax1 + yslope*(xloc-xdiff1)
          End if
          Do 20 j = 1,jmax(iblk)    ! Loop over no. of y points
            y(i,j) = (ystrt(iblk) + dy(iblk)*(j-1))*ymax
20          Continue
50        Continue
c
c-----Write the grid file in PLOT3D xyz format
c
      Write (2) ((x(i,j),i=1,imax(iblk)),j=1,jmax(iblk)),
&              ((y(i,j),i=1,imax(iblk)),j=1,jmax(iblk))
100      Continue
      Stop
      End

```

Figure 2 shows the resulting grid. Three zones are used, with grid sizes of  $17 \times 6$ ,  $33 \times 11$ , and  $17 \times 11$ , respectively.<sup>4</sup>



**Figure 2:** Test case grid

The grid file written by the above program, named *case4.xyz*, was converted to a *.cgd* file named *case4.cgd* using *cfcnv*, as shown in the following runstream. Lines in slanted type were typed by

<sup>4</sup>This geometry is simple enough that a single-zone grid would be sufficient, but a three-zone grid is used for illustrative purposes.



the user.

```

cfcnv
***** Common File Convert Utilities *****
CFCNVT - Version 1.41 (last changed 2000/06/13 15:48:02)

0: Exit program
2: Import   a Common File
3: Compress a Common File
4: Break Common File into multiple transfer files
5: Combine multiple transfer files into Common File
6: Append one Common File to another
7: Convert Common File binary to a text file
8: Convert Common File text to a binary file
11: Convert PLOT3D/Pegasus file to Common File
12: Convert GASP      file to Common File
13: Convert OVERFLOW file to Common File
14: Convert Common File to OVERFLOW file
15: Convert CFPOST GPU file to Common File GPC
16: Convert ascii rake to Common File rake CGF
17: Convert Pegasus 4.0 files to Common File

Enter the number from one of the above requests
11

PLOT3D file type menu

0: Main menu
1: Convert a PLOT3D Grid      (.x) file to CFS.
2: Convert a PLOT3D Solution (.q) file to CFS.

Enter the number from one of the above requests
1

PLOT3D Number of Grids menu

0: Main menu
1: PLOT3D Single zone format.
2: PLOT3D Multi  zone format.

Enter the number from one of the above requests
2

PLOT3D Zone dimension menu

0: Main menu
1: PLOT3D 2d zone format.
2: PLOT3D 3d zone format.

Enter the number from one of the above requests
1

```

## WIND User's Guide

### PLOT3D Format menu

- 0: Main menu
- 1: PLOT3D Formatted (ASCII).
- 2: PLOT3D Unformatted (sequential binary).

Enter the number from one of the above requests

2

### PLOT3D Iblank menu

- 0: Main menu
- 1: PLOT3D grid with IBLANK format.
- 2: PLOT3D grid without IBLANK format.

Enter the number from one of the above requests

2

### PLOT3D Precision menu

- 0: Main menu
- 1: PLOT3D Single precision format.
- 2: PLOT3D Double precision format.

Enter the number from one of the above requests

1

### PLOT3D INTOUT menu

- 0: Main menu
- 1: No INTOUT/XINTOUT file.
- 2: INTOUT file.
- 3: XINTOUT file.

Enter the number from one of the above requests

1

Enter PLOT3D .x file to convert with suffix

*case4.xyz*

Enter output Common File name with suffix

*case4.cgd*

Global maximums set as follows:

mimax	33
mjmax	11
mkmax	1
mpts:	363
recl:	726

Processing zone ZONE 1

Writing mesh data

Processing zone ZONE 2

Writing mesh data

```

Processing zone ZONE 3
Writing mesh data

0: Exit program
2: Import a Common File
3: Compress a Common File
4: Break Common File into multiple transfer files
5: Combine multiple transfer files into Common File
6: Append one Common File to another
7: Convert Common File binary to a text file
8: Convert Common File text to a binary file
11: Convert PLOT3D/Pegsus file to Common File
12: Convert GASP file to Common File
13: Convert OVERFLOW file to Common File
14: Convert Common File to OVERFLOW file
15: Convert CFPOST GPU file to Common File GPC
16: Convert ascii rake to Common File rake CGF
17: Convert Pegsus 4.0 files to Common File

Enter the number from one of the above requests
0

```

## 2.3 Define the Input

The next step is to define the input data. Input is required to specify the flow and initial conditions, the boundary conditions, and various parameters controlling the physical and numerical models to be used when running the code.

The primary file controlling how the WIND program is executed is the *Input Data (.dat)* file. With many CFD codes the input data are specified using Fortran namelist and/or formatted input. With WIND, the input is specified using descriptive keywords. The formatting rules for the *.dat* file are described in [Section 7.1](#).

This section is intended as an introduction to some of the more commonly-used keywords. After reading the information presented here, a new user should supplement it with the detailed information in [Section 10](#). For many cases, the default values for the various keyword options are acceptable, but users should become familiar with all of the options for the most effective use of the WIND code.

### 2.3.1 Descriptive Header and Comments

The first three lines of the file are reserved for geometry, flow condition, and arbitrary titles, respectively. Each of these titles may be up to 64 characters long.

[Comment lines](#), beginning with a `/`, may be placed anywhere in the file after the first three lines. The readability of the *.dat* file may be improved significantly through the liberal use of comments — for example, to separate logical sections of the data file like boundary conditions, numerical operators, convergence monitoring parameters, etc.

### 2.3.2 Boundary Conditions

With most CFD codes, boundary conditions are completely specified in the input data file. With WIND, however, setting boundary conditions is a two-step process, defining first the *type* of boundary, and then any *values* that are needed.

The first step is to label each boundary of each zone with the *type* of boundary condition to use, such as “viscous wall,” “outflow,” or “coupled.” This is done using the GMAN pre-processing code, and the information is stored in the Common Grid (.*cgd*) file. Details on the boundary condition types available for use with WIND are in [Section 5](#) and in the *GMAN User's Guide*.

Boundary condition types may be specified for all or part of a boundary, allowing multiple boundary conditions on a single boundary.

“Coupled” zonal interface boundaries do not have to be explicitly labeled by the user. GMAN can automatically examine the grid to find them and determine the zones involved, compute the geometric interpolation factors, and store the information in the .*cgd* file. GMAN is also used to cut holes and generate interpolation coefficients for overlapping (“chimera”) boundaries. The process is currently not completely automated for chimera boundaries.

The second step is to define any *values* needed for a particular boundary condition, such as an exit pressure or a bleed rate. This information is specified in the .*dat* file. Keywords are available to specify conditions at inflow boundaries ([ARBITRARY INFLOW](#)), at outflow boundaries ([COMPRESSOR FACE](#), [DOWNSTREAM MACH](#), [DOWNSTREAM PRESSURE](#), [MASS FLOW](#)), along solid walls ([MOVING WALL](#), [TTSPEC](#), [WALL TEMPERATURE](#)), in bleed and blowing regions ([BLEED](#), [BLOW](#)), and across actuators and screens ([ACTUATOR](#) | [SCREEN](#)).

### 2.3.3 Flow and Initial Conditions

The flow conditions (Mach number, and static or total pressure and temperature, plus the angles of attack and yaw) are specified using the [FREESTREAM](#) keyword. These conditions, along with a reference length based on the units used in the .*cgd* file, are used as the reference conditions and determine the Reynolds number.

The usual procedure with WIND is to start a new problem by setting the initial conditions at each grid point equal to the values specified using the [FREESTREAM](#) keyword. Other keywords allow different values to be used in different zones ([ARBITRARY INFLOW](#)), a boundary layer to be added along a specified surface in a zone ([BL\\_INIT](#)), and reinitialization of the flow in specified zones after a restart ([REINITIALIZE](#)). Previously-run, partially-converged cases will normally be restarted using the current solution as initial conditions ([RESTART](#)). More information about flowfield initialization may be found in [Section 3.5](#).

### 2.3.4 Physical Model Controls

**Dimensionality.** WIND may be used for three-dimensional, two-dimensional, quasi-three-dimensional, or axisymmetric configurations. Internally, the WIND code treats the grid as three-dimensional, with indices  $i$ ,  $j$ , and  $k$ . Two-dimensional cases simply have  $k_{max} = 1$ , with the  $i$ - $j$  grid lying in a non-zero, constant  $z$  plane. The effect of area variation in an otherwise two-dimensional configuration may be modeled using WIND's quasi-three-dimensional capability, which is activated by setting the  $z$ -coordinate equal to the “width” of the geometry at each grid point. Axisymmetric configurations are modeled using a two-dimensional grid in conjunction with the [AXISYMMETRIC](#) keyword. More details may be found in [Section 3.1](#).

**Flow Equations.** The default equations solved by the WIND code are the full Reynolds-averaged Navier-Stokes equations. Keywords are available to solve the Euler equations ([TURBULENCE](#)), parabolized Navier-Stokes equations ([MARCHING](#), [IMPLICIT](#)), or the thin-layer Navier-Stokes equations ([TSL](#)). See [Section 4.2.2](#) for more information about the thin-layer option.

**Turbulence Model.** A variety of turbulence models are available in the WIND code through the [TURBULENCE](#) keyword. These include the Baldwin-Lomax, Cebeci-Smith, and P. D. Thomas algebraic models; the Spalart-Allmaras and Baldwin-Barth one-equation models; and the Chien  $k$ - $\epsilon$  and Menter Shear Stress Transport (SST) two-equation models.<sup>5</sup> A laminar flow option is also available using the [TURBULENCE](#) keyword, and a laminar-turbulent transition region may be modeled using the [TTSPEC](#) keyword. See [Section 3.3](#) for more information.

**Gas Model and Chemistry.** A variety of gas models are available in WIND to complete the equation set. The fluid may be treated as a thermally and calorically perfect gas, a thermally perfect gas (frozen chemistry), equilibrium air, or a mixture undergoing a finite rate chemical reaction. For a thermally and calorically perfect gas, the values of  $\gamma$ , the laminar and turbulent Prandtl numbers, and the gas constant may be set using the [GAS](#) keyword. Real gas chemistry models are selected using the [CHEMISTRY](#) keyword. Several different chemistry packages are available in the form of files containing thermodynamic data, finite rate coefficients, and transport property data, described in [Section 7.10](#).

### 2.3.5 Numerical Model Controls

**Time Stepping.** In WIND, the number of iterations or time steps to perform in a given run is defined in terms of *cycles* and *iterations per cycle*. An iteration advances the solution one time step. A cycle consists of a solution pass over all the zones. Zone coupling, the process whereby WIND exchanges flowfield information between zones, only occurs at the end of each cycle. The Common Flow (*.cfl*) file is also updated only at the end of each cycle. The number of cycles to be performed is set using the [CYCLES](#) keyword, and the number of iterations per cycle, which may vary from zone to zone, is set using the [ITERATIONS](#) keyword. The default is five iterations per cycle.

The time step size is controlled by the [CFL#](#) keyword. By default, local time stepping is used, so that the solution advances at a different rate at each grid point. For unsteady problems, of course, the same time step size should be used throughout the flowfield, and an option is available with the [CFL#](#) keyword for this purpose. A Runge-Kutta time step formulation is also available, using the [STAGES](#) keyword, and may be used for both steady and unsteady flows.

See [Section 4.1](#) for more information about cycles and iterations, and [Section 4.5](#) for more about time step options.

**Implicit Operator.** The [IMPLICIT](#) keyword allows a variety of implicit operators to be specified, including point Jacobi, Gauss-Seidel, and MacCormack modified approximate factorization. Also available are options to: (1) turn off the implicit operator completely, resulting in an explicit calculation; (2) use a scalar (diagonalized) implicit operator; or (3) use a full block implicit operator. For these last three options, a different implicit operator may be specified for each computational direction. This keyword is also used when the parabolized Navier-Stokes equations are being solved to specify that the PNS implicit operator should be used in the  $\xi$  direction.

---

<sup>5</sup>It should be noted that the algebraic models are older and infrequently used. They may not work as well as in some other codes, and there may be bugs in their implementation.

The default is to use the full block operator in viscous directions, and the scalar (diagonalized) operator in inviscid directions.

The **IMPLICIT BOUNDARY** keyword may be used to specify that implicit boundary conditions are to be used on “wall” boundaries. This should improve stability when the CFL number is above about 1.3.

**Explicit Operator.** Through use of the **RHS** keyword, a wide variety of explicit operators are available for evaluation of the first-derivative terms on the right-hand side. These include a central difference scheme, upwind schemes due to Coakley and Roe, and a modified version of the Roe scheme for stretched grids. Depending on the type of scheme used, the accuracy may be specified as anywhere from first to fifth order. The default is Roe’s second-order flux-difference splitting algorithm, modified for stretched grids.

**Damping Schemes.** Various smoothing options are available in WIND to dampen instabilities that may occur under certain conditions. These include second- and fourth-order explicit smoothing (**SMOOTHING**, **BOUNDARY-DAMP**), and total variation diminishing (TVD) flux limiting for some of the explicit operators (**TVD**, **BOUNDARY TVD**). More details on the various damping options are in [Section 4.3](#).

**Convergence Acceleration.** The **ACCELERATE** keyword may be used, in conjunction with the **SMOOTHING** and **CFL#** keywords, to increase the time step near the beginning of a calculation, in order to more quickly get through the start-up transients that may occur in the first few hundred iterations.

A grid sequencing capability is also available, using the **SEQUENCE** keyword, that may help speed convergence. With this option, grid points are removed from selected regions of the flowfield, resulting in a coarse-grid solution which is obtained in a fraction of the time it would have taken for a full-grid solution. At the end of each run, the solution is interpolated back onto the original grid to aid in restarting the solution, and to provide a continuous flowfield for post-processing. The full grid should of course be used when the solution nears convergence.

See [Section 4.6](#) for more information about the **ACCELERATE** option, and [Section 4.2.1](#) for more about the grid sequencing capability.

**Convergence Monitoring Parameters.** The convergence criterion, in terms of the required value or reduction of the maximum residual, may be specified using the **CONVERGE** keyword. Integrated forces, moments, and/or mass flow may also be used to monitor convergence, by using the **LOADS** keyword to periodically compute and print these values for specified three-dimensional regions and/or two-dimensional areas of a computational surface.

For unsteady flow problems, where the time step is being specified in seconds and is the same throughout the flow field, a time history tracking capability is also available using the **HISTORY** keyword. Selected flow variables may be computed at specified grid points, and written to a separate *Time History (.cth)* file.<sup>6</sup>

More information about monitoring convergence is presented in [Section 2.5](#) of this tutorial, and in [Section 6](#).

---

<sup>6</sup>However, the time history tracking capability does not currently work when WIND is run in parallel mode, and will cause an abort.

### 2.3.6 Test Case 4 Input

**Boundary Condition Types.** *GMAN* was used in graphical mode to set the boundary condition types and store the information in the *.cgd* file. The interface boundaries between the three zones were automatically identified, and the geometric interpolation factors were computed and stored in the *.cgd* file. The inflow boundary ( $i = 1$  in zones 1 and 2) was labeled as “arbitrary inflow,” the outflow boundary ( $i = i_{max}$  in zone 3) was labeled as “outflow,” and the top and bottom boundaries ( $j = 1$  in zones 1 and 3, and  $j = j_{max}$  in zones 2 and 3) were labeled as “inviscid wall.”

The first step, obviously, is to start GMAN.

```
gman
```

```
***** gman *****
```

```
Select the desired version from the following list.
```

- ```
0) END
1) gman_pre optimized version
```

```
Enter number of executable.....[1]:
```

```
GMANPRE - Version 6.134 (last changed 1999/09/16 14:33:33)
Creating journal file 'gman.jou'.
```

```
Enter SWITCH or GRAPHICS to change to graphics mode.
```

```
GMAN:
```

At this point, you may enter commands individually at the **GMAN:** prompt. Or, you could enter **SWITCH** or **GRAPHICS** to enter graphics mode.

The rest of this section describes in detail the use of GMAN for the tutorial test case. The graphics mode steps are on the left, with the Main Menu steps left-aligned and the Menu Options indented. Most of these are accomplished in GMAN by clicking on the listed menu item using the left mouse button. A few require entering text in the prompt area at the bottom of the screen. (See the “Graphical User Interface Basics” section of the *GMAN User’s Guide* for a description of the various sections of the GMAN screen layout.)

The command line equivalents are shown on the right. Note that, in general, several graphics mode steps become consolidated into a single command.

We first need to tell GMAN the name of the file containing the grid.

#### Graphics Mode

```
Common File
  enter case4.cgd
```

#### Command Line Mode

```
file case4.cgd
```

Next, we use GMAN’s automated procedure to define zonal coupling information.

### Graphics Mode

```
BOUNDARY COND.
AUTO COUPLE
  RUN AUTO COUP
```

### Command Line Mode

```
automatic couple face zone all
```

Next, for zone 1, we define the inflow and lower wall boundaries.

### Graphics Mode

```
PICK ZONE/BN DY
  1:  (from Zone List)
  I1
MODIFY BNDY
CHANGE ALL
  ARBITRARY INFLO
BOUNDARY COND.
  YES - UPDATE FILE
```

### Command Line Mode

```
zone 1
boundary i1
arbitrary inflow
update
```

```
PICK ZONE/BN DY
  J1
MODIFY BNDY
CHANGE ALL
  INVISCID WALL
BOUNDARY COND.
  YES - UPDATE FILE
```

```
boundary j1
inviscid wall
update
```

For zone 2, we define the inflow and upper wall boundaries.

### Graphics Mode

```
PICK ZONE/BN DY
  2:  (from Zone List)
  I1
MODIFY BNDY
CHANGE ALL
  ARBITRARY INFLO
BOUNDARY COND.
  YES - UPDATE FILE
```

### Command Line Mode

```
zone 2
boundary i1
arbitrary inflow
update
```

```
PICK ZONE/BN DY
  JMAX
MODIFY BNDY
CHANGE ALL
  INVISCID WALL
BOUNDARY COND.
  YES - UPDATE FILE
```

```
boundary jmax
inviscid wall
update
```



And for zone 3, we define the outflow and both wall boundaries.

| <u>Graphics Mode</u>         | <u>Command Line Mode</u> |
|------------------------------|--------------------------|
| PICK ZONE/BNDY               | <b>zone</b> 3            |
| 3: ( <i>from Zone List</i> ) |                          |
| IMAX                         | <b>boundary</b> imax     |
| MODIFY BNDY                  | outflow                  |
| CHANGE ALL                   |                          |
| OUTFLOW                      |                          |
| BOUNDARY COND.               | <b>update</b>            |
| YES - UPDATE FILE            |                          |
| PICK ZONE/BNDY               |                          |
| j1                           | <b>boundary</b> j1       |
| MODIFY BNDY                  | inviscid wall            |
| CHANGE ALL                   |                          |
| INVISCID WALL                |                          |
| BOUNDARY COND.               | <b>update</b>            |
| YES - UPDATE FILE            |                          |
| PICK ZONE/BNDY               |                          |
| JMAX                         | <b>boundary</b> jmax     |
| MODIFY BNDY                  | inviscid wall            |
| CHANGE ALL                   |                          |
| INVISCID WALL                |                          |
| BOUNDARY COND.               | <b>update</b>            |
| YES - UPDATE FILE            |                          |

Finally, it's a good idea to check the boundary conditions to make sure all is OK.

| <u>Graphics Mode</u>          | <u>Command Line Mode</u> |
|-------------------------------|--------------------------|
| TOP                           |                          |
| CHECK                         | <b>zone</b> all          |
| CHECK BOUNDARY                | <b>check boundary</b>    |
| PICK ZONE                     |                          |
| ALL ( <i>from Zone List</i> ) |                          |
| RUN BNDY CHKS                 |                          |

After hitting the **Enter** key to return to GRAPHICS mode, we can quit GMAN.

| <u>Graphics Mode</u> | <u>Command Line Mode</u> |
|----------------------|--------------------------|
| END                  | <b>exit</b>              |
| YES - TERMINATE      |                          |

**Input Data (.dat) File.** The Input Data File for Test Case 4, named *case4.dat*, is listed below. The explanatory notes, in italics, are not part of the file.

|                                        |                                                                                                                                         |
|----------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| WIND test case 4, 2-D, 3 zones         | <i>Titles</i>                                                                                                                           |
| Subsonic internal flow                 |                                                                                                                                         |
| Run 1                                  |                                                                                                                                         |
|                                        |                                                                                                                                         |
| / Inlet conditions                     |                                                                                                                                         |
| Freestream total 0.78 15.0 600.0 0. 0. | <i>Inlet <math>M</math>, <math>p_t</math> (psi), <math>T_t</math> (<math>^{\circ}R</math>), <math>\alpha</math>, <math>\beta</math></i> |
|                                        |                                                                                                                                         |
| / Boundary conditions                  |                                                                                                                                         |
| Downstream pressure 14.13 zone 3       | <i>Exit <math>p</math> (psi)</i>                                                                                                        |
|                                        |                                                                                                                                         |
| / Numerics                             |                                                                                                                                         |
| Cycles 1000                            | <i>Run 1000 cycles</i>                                                                                                                  |
|                                        |                                                                                                                                         |
| / Viscous terms                        |                                                                                                                                         |
| Turbulence euler                       | <i>Solve inviscid equations</i>                                                                                                         |
|                                        |                                                                                                                                         |
| / Convergence data                     |                                                                                                                                         |
| Loads                                  | <i>Compute <math>m</math> every 5 iterations at:</i>                                                                                    |
| print planes frequency 5               |                                                                                                                                         |
| zone 1                                 |                                                                                                                                         |
| surface i 1 mass                       | <i>zone 1 entrance</i>                                                                                                                  |
| zone 2                                 |                                                                                                                                         |
| surface i 1 mass                       | <i>zone 2 entrance</i>                                                                                                                  |
| zone 3                                 |                                                                                                                                         |
| surface i 1 mass                       | <i>zone 3 entrance</i>                                                                                                                  |
| surface i last mass                    | <i>zone 3 exit</i>                                                                                                                      |
| Endloads                               |                                                                                                                                         |

## 2.4 Run the Code

### 2.4.1 The *wind* Script

WIND is invoked using a Unix script, called *wind*, which prompts for the executable to be used (since production and beta versions of WIND may both be available on a system), the names of the various input and output files (which should be entered *without* the three-letter suffix), and for the queue in which the job is to run. If a *multi-processing control* (.mpc) file is present with the same base name as the .dat file (see [Section 2.4.2](#)), it also issues a prompt to verify that the job is to be run in parallel mode. The script then links the files to the appropriate Fortran units, and either starts WIND interactively or submits the job to the specified “at” or “batch” queue. Details on the *wind* script are in [Section 8.1](#).

There are a couple of very convenient features built into the *wind* script. The first allows a run to be stopped at (or more exactly, shortly after) a pre-determined time through the use of an *NDSTOP* file. This is useful when an overnight run must be stopped before morning, when the workstations being used will be needed for interactive work. The second allows the user to break a long run into “sub-runs,” by writing a script called *wind\_post* containing tasks to perform between each run. This is useful, for example, when the complete solution is to be saved at various time intervals in an unsteady problem. Details on the use of the *NDSTOP* file and the *wind\_post* script

are in [Section 7.8](#) and [Section 8.2](#), respectively.

### 2.4.2 Parallel Operation

When WIND is run in parallel mode, multiple systems connected via a network work together as though they were a single computer. These systems are typically workstation class machines and need not be all from the same vendor.

A master-worker approach is used. Grid zones are distributed from the master system to the worker systems for processing. (Note that the master may also be a worker.) Each zone is solved in parallel with other zones on other systems. The systems exchange boundary information at the end of each solution cycle to propagate information throughout the flowfield. If there are fewer workers than zones, a worker will be assigned another zone when it finishes its current assignment.

The user specifies the names of the participating worker systems via a *multi-processing control* (.mpc) file, which must have the same base name as the .dat file. The user must of course have accounts on the master and worker systems, and the master must have remote shell access to each of the workers, via a *hosts.equiv* or *.rhosts* file.<sup>7</sup> This is all that is required to utilize the distributed parallel processing capability of WIND. The PVM software needed for parallel operation, and the WIND code itself, will be copied from the master to temporary directories on the workers.

Additional details about running WIND in parallel mode may be found in [Section 9](#).

### 2.4.3 Running Test Case 4

To run Test Case 4, simply issue the *wind* command, and respond to the prompts as appropriate. The following terminal session shows how the case was run as a serial batch job on a Unix workstation. Lines in slanted type were typed by the user.

```
% wind -runinplace
Running command line version of WIND.

***** WIND Run Script *****

Current wind settings are:

--Wind test mode set to NODEBUG
--Wind debugger set to DEFAULT
--Wind run que set to PROMPT
--Wind run in place mode is set to YES
--Wind multi-processor mode set to NO
--Wind run directory set to PROMPT
--Wind bin directory set to /usr2/wind/wind

Select the desired version

0: Exit wind
1: alpha version
```

---

<sup>7</sup> If *ssh* (secure shell) is being used, the corresponding files are *shosts.equiv* and *.shosts*.

## WIND User's Guide

```
2: Version 2.0
3: Version 3.0
4: Version 4.0

Enter number or name of executable.....[4]:
Basic input data.....(*.dat): case4
Output data.....(*.lis,<CR>=case4):
Mesh file.....(*.cgd,<CR>=case4):
Flow data file.....(*.cfl,<CR>=case4):

*****
case4.cfl does not exist, a fresh start will be performed.
*****

Enter a queue number from the following list or <CR> for default:
1: REAL (interactive)
2: AT_QUE
3: QSUB_QUE
4: QSUB_PBS_QUE

Queue name.....(<CR> for 1): 2

Enter at que (a,b..).....(<CR> for default):
Deferred start time.....(hhmm [day],<CR>=now):
Set stop flag at.....(hhmm [day],<CR>=don't set):

Version.....: /usr2/wind/wind/SGI6.5/R10000/wind4.exe
Input file name.....: case4.dat
Wind output to.....: case4.lis
Grid file name.....: case4.cgd
Flow file name.....: case4.cfl
Job run que type is...: AT_QUE

Press <cr> to submit job, another key (except space) and <cr> to abort:
warning: commands will be executed using /bin/sh
job 888417781.b at Wed Apr 18 12:53:43 2001
All done, have a nice day!!
```

There are several points to note from this terminal session.

- This case was run using the *-runinplace* option to the *wind* script, which means that the WIND code will be run in the current directory, and that output files will be written in the current directory.

The default is to run in a different (i.e., remote) directory, and is intended primarily for use with NFS-mounted home directories. In that case, it's faster to write the output files into a scratch directory on the system used to run the WIND code, rather than into the NFS-mounted home directory. The output files are automatically copied to the current directory at the end of the job.

Note that the terminology here is unfortunately a bit confusing. With an NFS-mounted home

directory, running remotely really means running on a system different from the one the home directory is on. The “remote” system may actually be the local system originally logged onto.

If this case were run without the *-runinplace* option, the user would be prompted to enter the root name of the remote run directory, as follows:

```
# Note the remote directory is assumed to exist on remote host #
Enter the remote run root directory...(CR> for /tmp):
```

The full name of the remote run directory will be *rootname/userid/basename.scr*, where *rootname* is your response to the above prompt, *userid* is your userid, and *basename* is the base name of your *.dat* file. The default of */tmp* for the root name implies that, generally, the “remote” system is actually the one the user logged into. It also means that, if you aren’t using NFS-mounted home directories, and you forget to add the *-runinplace* option, no real harm is done. The output will be created under */tmp*, then copied to the current directory when the job finishes.

- The default for the base name of the *.lis*, *.cgd*, and *.cfl* files is the same as that entered for the *.dat* file, and the three-letter suffixes *should not* be entered.
- A “fresh start” is being done for this case, since the *.cfl* file does not exist. If the *.cfl* file exists, WIND will automatically restart from the existing flow field.
- To run the job interactively, choose the **REAL** queue. This is intended mostly for debugging short runs, or checking that a case will successfully start. The user will be asked if the output should be written to the screen instead of the *.lis* file.

The **AT\_QUE** queue is used to run a batch job using the Unix *at* or *batch* command. If the response to the **Deferred start time** prompt is defaulted, the job will be started immediately using the *batch* command. Any other response will result in the *at* command being used to start the job at the specified time.<sup>8</sup>

The **QSUB** queue choices shown above are intended for use on systems with NQS (Network Queueing System) software installed. The user will be prompted for the necessary NQS queue name, start time, etc. The **QSUB** choices will not appear on systems without a *qsub* command.

- As noted earlier, if a multi-processing control (*.mpc*) file is present, the user is also asked to verify that the job is to be run in parallel mode, as follows:

```
A multiprocessing control file exists....
Do you want to run in multi-processor mode (y/n,<CR>=y):
```

## 2.5 Monitor Convergence

For complex real-world applications, it is generally not feasible to expect a converged solution in a single run. The times required to achieve convergence are generally too long and problems may occur which could corrupt the solution. Thus, executing the code several times and restarting from the previous solution is often the best approach. If problems do occur, the input parameters can be adjusted without starting from scratch.

Monitoring and properly assessing convergence levels during a WIND run (as well as examining the flowfield itself, as discussed in [Section 2.6](#)) are thus crucial in obtaining meaningful, useful results. WIND users may track convergence by following residuals and/or integrated forces, moments, and

<sup>8</sup>*Do not* type **now** in response to the prompt. That will result in an “at” job being submitted, and a “too late” error.

mass flow via the **LOADS** keyword. For engineering applications, the recommended convergence monitoring method is the tracking of integrated quantities of interest. For example, if a wing/body geometry is being modeled to determine drag, the integrated drag should be monitored and some reasonable bounds on drag oscillations should be used as the convergence criterion.

The solution residuals are included in the *List Output* (.lis) file. For each zone, WIND prints the zone number, cycle number, location of the maximum residual, equation number for which the maximum residual occurred, the value of the maximum residual, and the  $L_2$ -norm of all the residuals for all the equations over all the points in the zone. By default, the residuals are printed each iteration. The output interval may be changed, however, using the **CYCLES** and **ITERATIONS** keywords.

The integrated parameters that are chosen in the Input Data file via the **LOADS** keyword will also be listed in the .lis file. The integration may be done over a number of specified three-dimensional regions and/or two-dimensional areas of a computational surface.

For unsteady flow problems, where the time step is being specified in seconds and is the same throughout the flow field, a time history tracking capability may be used. Computed values of selected variables at specified grid points may be periodically written to a separate *Time History* (.cth) file. This capability is activated using the **HISTORY** keyword. Currently, the only values that may be tracked with this option are Mach number, static pressure, static temperature, and the three Cartesian velocity components  $u$ ,  $v$ , and  $w$ .<sup>9</sup>

A utility included with the WIND code called *resplt* can be used to extract the residuals and/or integrated quantities from the .lis file, and create a GENPLOT file<sup>10</sup> for post-processing.

An analogous utility called *timplt* can be used for the values stored in the .cth file.

Additional information about the various methods for monitoring convergence is presented in [Section 6](#).

### 2.5.1 Test Case 4 Convergence

*resplt* was used to extract the maximum residual and the  $L_2$ -norm of the residuals from the List Output (.lis) file and create GENPLOT files. As an example of the use of *resplt*, the following terminal session shows how a GENPLOT file containing the maximum residual was created. A GENPLOT file containing the  $L_2$ -norm of the residual was created similarly, using selection “2”. Lines in slanted type were typed by the user.

```
% resplt
Enter full name of output list file:
case4.lis

Exit                0
Select Zone(s)      91
Select Frequency     92
Confined Outflow
  Mass Flow Ratio    15
  Back Pressure      16
  Average p0         93
Residuals    Big   L2      Integ. Planes Zone Grand
```

<sup>9</sup>As noted earlier, however, the time history tracking capability does not currently work when WIND is run in parallel mode, and will cause an abort.

<sup>10</sup>The format of a GENPLOT file is described in the *CFPOST User's Guide*.

|           |    |    |          |    |    |    |
|-----------|----|----|----------|----|----|----|
| NS        | 1  | 2  | Force    | 11 | 5  | 8  |
| k-e       | 3  | 4  | Lift     | 17 | 18 | 19 |
| B-B       | 20 | 21 | Moment   | 12 | 6  | 9  |
| S-A       | 22 | 23 | Momentum | 13 | 7  | 10 |
| SST       | 24 | 25 | Mass     | 14 | 26 | -  |
| NEWTON NS | 51 | 52 |          |    |    |    |

Enter Selection

1

Reading residual data...

6550 data points read from list file.

Sorting residual data...

Sorting complete.

Enter FULL name of genplot file:

*chist.big.gen*

|                  |     |    |               |      |       |    |
|------------------|-----|----|---------------|------|-------|----|
| Exit             | 0   |    |               |      |       |    |
| Select Zone(s)   | 91  |    |               |      |       |    |
| Select Frequency | 92  |    |               |      |       |    |
| Confined Outflow |     |    |               |      |       |    |
| Mass Flow Ratio  | 15  |    |               |      |       |    |
| Back Pressure    | 16  |    |               |      |       |    |
| Average p0       | 93  |    |               |      |       |    |
| Residuals        | Big | L2 | Integ. Planes | Zone | Grand |    |
| NS               | 1   | 2  | Force         | 11   | 5     | 8  |
| k-e              | 3   | 4  | Lift          | 17   | 18    | 19 |
| B-B              | 20  | 21 | Moment        | 12   | 6     | 9  |
| S-A              | 22  | 23 | Momentum      | 13   | 7     | 10 |
| SST              | 24  | 25 | Mass          | 14   | 26    | -  |
| NEWTON NS        | 51  | 52 |               |      |       |    |

Enter Selection

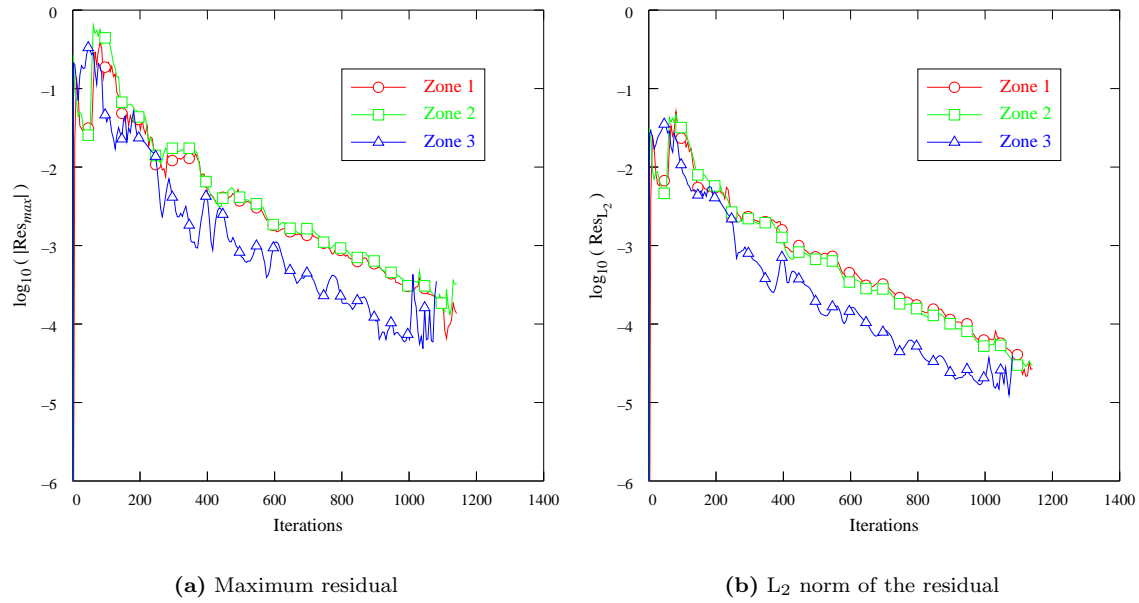
0

The convergence history for Test Case 4, in terms of the maximum residual and the  $L_2$  norm of the residual, is shown in [Figure 3](#).

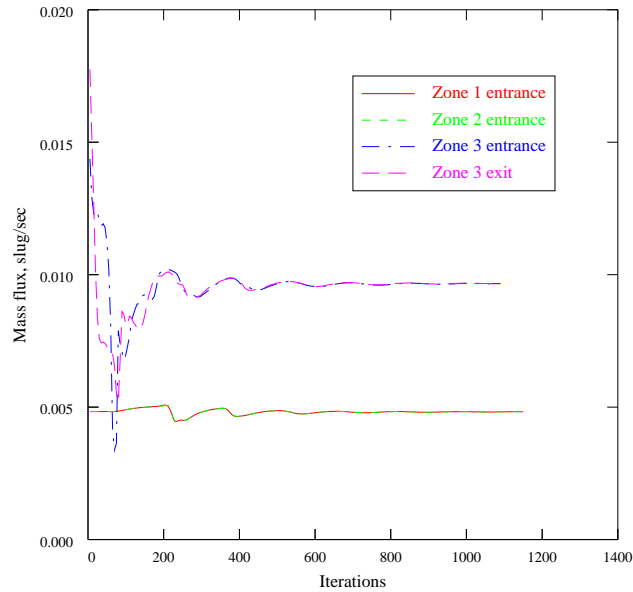
As noted above, it's usually more meaningful to use a physical quantity of interest when monitoring convergence. For this test case, one of the physical quantities of interest is the mass flow rate.

The Input Data file for this case specified that the integrated mass flux was to be computed at the entrances to all three zones, and at the exit of zone 3. GENPLOT files containing these integrated values were created using *resplt*, as illustrated above, using selection "14".

The time history for these parameters is shown in [Figure 4](#), where the computed mass flux is plotted as a function of iteration number at the entrances to all three zones, and at the exit of the duct.



**Figure 3:** Test case convergence history in terms of residuals



**Figure 4:** Test case convergence history in terms of mass flux

Because this test case is a simple 2-D inviscid flow, with a coarse mesh, it converged very quickly in a single run. The various convergence parameters were thus examined only at the end of the run. When running a more realistic, real-world configuration, convergence parameters like those shown in [Figure 3](#) and [Figure 4](#) should be examined at the end of each run.



## 2.6 Examine the Results

Of course, the purpose of the solution process is to determine the features of the flow which can help answer the questions that drove the decision to perform the simulation in the first place. And, as indicated in the previous section, it's important to periodically examine the computed results during a run to help assess convergence and detect numerical problems that might be corrected by adjusting the input.

There are two types of information that can be extracted from the flow simulation — specific quantitative data and qualitative patterns. The first type includes things like pressure distributions, drag, and total flow rate. The second type includes, for example, 2-D slices or full 3-D visualization of pressure contours, streamlines, or isothermal surfaces.

### 2.6.1 Using CFPOST

All flowfield results computed by WIND, including the mean flow variables, turbulence model variables, and chemistry variables, are written into a Common File called a *Common Flow* (.cfl) file.<sup>11</sup> The **CFPOST** utility, included with the WIND distribution, is a post-processing tool for examining the contents of the .cfl file.

With CFPOST a wide variety of variables and integrated values may be computed. Listings of quantitative results may be sent to the screen or to a file. PLOT3D files may be created for other plotting packages and post-processors to use in displaying qualitative results. CFPOST can also be used to create *x-y*, contour, and vector plots directly, with PostScript output. Commands are available to precisely specify the information of interest, the domain of interest, and the units in which the results are to be presented. More details may be found in the *CFPOST User's Guide*.

### 2.6.2 Test Case 4 Results

The desired results from this calculation were the mass flow rate and the static pressure distribution. The mass flow rate is available in the List Output (.lis) file, as output generated via the LOADS keyword, and also from the GENPLOT files created while monitoring convergence. The result was  $9.7 \times 10^{-3}$  slug/sec.

To examine the static pressure distribution, CFPOST was first used to create a PLOT3D q file called *case4.q* from the Common Flow file *case4.cfl*, as follows:<sup>12</sup>

```
% cfpост

***** cfpост *****

Select the desired version from the following list.

0) END
1) cfpост_pre optimized version

Enter number of executable.....[1]:
```

<sup>11</sup> As noted previously, WIND also supports the use of CGNS files for the grid and flow solution, using the **CGNSBASE** keyword.

<sup>12</sup>CFPOST can also be used to create a PLOT3D xyz file, but as noted in [Section 2.2.3](#), an xyz file for this configuration was created during the mesh generation process.

```

CFPOST - Version 3.129 (last changed 1999/07/13 20:46:07)
CFPOST> solution case4.cfl
1WIND test case 4, 2-D, 3 zones
Subsonic internal flow

Freestream conditions (* indicates calculated)

Mach number                0.780
Angle of attack             0.000 degrees
Yaw angle                  0.000 degrees
Gamma                      1.400
Gas constant (R)           286.959 m2/s2-K      1716.000 ft2/s2-R
Static pressure             69194.1 N/m2        10.0357 lbf/in2
Static temperature         297.173 K           534.912 R
Static density*            0.811408 kg/m3       0.157439E-02 slug/ft3
Total pressure*            103421. N/m2        15.0000 lbf/in2
Total temperature*        333.333 K           600.000 R
Total density*            1.08121 kg/m3       0.209790E-02 slug/ft3
Speed of sound             345.524 m/s        1133.61 ft/s
Velocity                   269.509 m/s        884.216 ft/s
Dynamic pressure           29468.4 N/m2        4.27402 lbf/in2
Laminar Viscosity          0.183722E-04 kg/m-s   0.383710E-06 slug/ft-s
Enthalpy*                 298468. m2/s2       0.321268E+07 ft2/s2
Stagnation Enthalpy*      334785. m2/s2       0.360360E+07 ft2/s2
CFPOST> subset i all j all k all
CFPOST> zone 1
CFPOST> zone 2
CFPOST> zone 3
CFPOST> plot3d q case4.q 2d mgrid unformatted
Processing ZONE 1...
Processing ZONE 2...
Processing ZONE 3...
CFPOST> quit

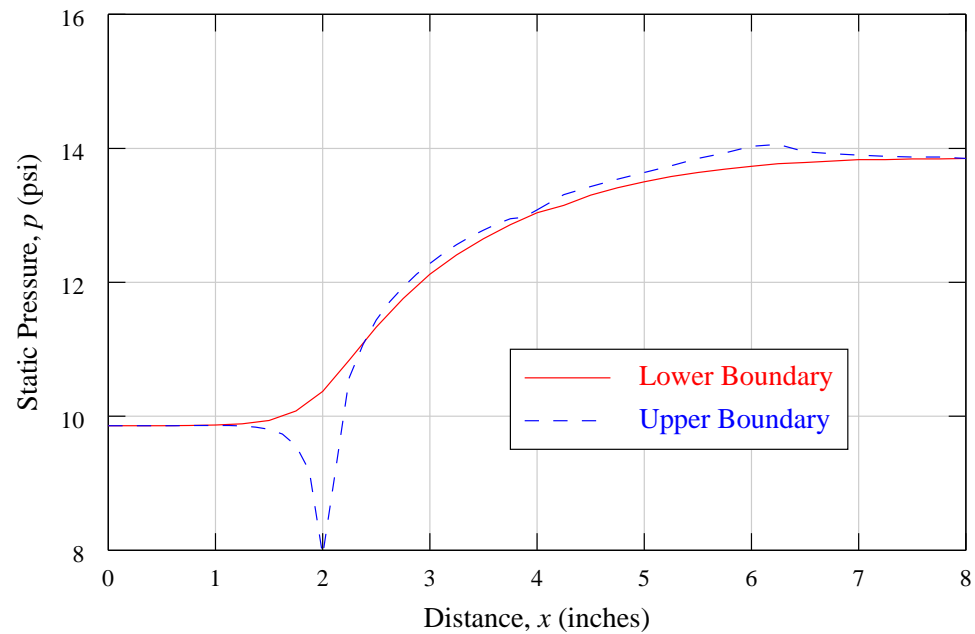
```

The PLOT3D xyz and q files were then used as input to other post-processing packages to examine the computed results. The computed static pressure distribution along the upper and lower walls is shown in [Figure 5](#), and the static pressure field is shown in [Figure 6](#).

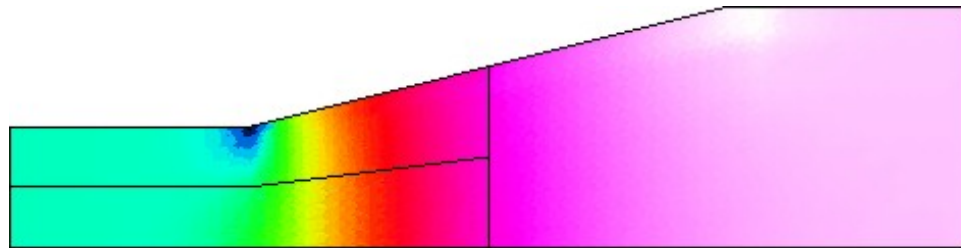
## 2.7 Summary

The steps in the generalized solution process listed earlier may be expanded and restated specifically for the WIND code as follows:

- Gather information detailed enough to specify the problem within the required accuracy, including the geometry, flow conditions, and boundary conditions.
- Create a grid file using any convenient grid generation software, saving the file in PLOT3D xyz format.
- Convert the PLOT3D xyz file to a Common Grid (.cgd) file using *cfcvnt*.
- Store the boundary condition types and zonal connectivity data in the .cgd file using *GMAN*.



**Figure 5:** Test case static pressure distribution



**Test case constant static pressure contours**

**Figure 6:** Test case static pressure field

- Prepare the Input Data (*.dat*) file, defining boundary condition values, initial conditions, program control parameters, and integrated parameters for monitoring convergence.
- For parallel execution, prepare the *multi-processing control* (*.mpc*) file.
- Run the WIND code using the *wind* script supplied with the code.
- Monitor convergence by examining the residuals and integrated values in the List Output (*.lis*) file. The utilities *resplt* and **CFPOST** may be helpful.
- Periodically examine the computed results in the Common Flow (*.cfl*) file using CFPOST, creating PLOT3D files for other post-processing packages if desired.



## 3 Geometry and Flow Physics Modeling

### 3.1 Symmetry Considerations

WIND may be used for axisymmetric, two-dimensional, or three-dimensional geometric configurations. Two-dimensional grids may be used not only for two-dimensional cases, but also for axisymmetric and area variation (quasi-three-dimensional) cases.

#### 3.1.1 Three-Dimensional Cases

In three dimensions, each zone’s computational mesh is comprised of six boundary faces and an interior grid, the points of which are identified by three indices, usually labeled  $(i, j, k)$ . Boundary conditions for each of the six faces must be specified with GMAN before the grid may be used with WIND.

#### 3.1.2 Two-Dimensional Cases

In two dimensions, the grid must be oriented such that the maximum  $k$ -index of the grid is 1. In other words, a two-dimensional grid is defined by four boundary faces and an interior grid labeled by  $i$ - and  $j$ -indices. The grid must also reside in a non-zero, constant  $z$ -coordinate plane. The actual value used will not affect solution convergence or flowfield features, but it will affect flux-related post-processing calculations such as mass flow. For this reason, a value of 1.0 is recommended. Boundary conditions for the four boundary lines must be specified in GMAN.

#### 3.1.3 Area Variation (Quasi-3D) Cases

The effect of area variation on two-dimensional computational models may be computed by using WIND’s “quasi-three-dimensional” capability, which is activated simply through changes in the  $z$ -coordinate. The value of the  $z$ -coordinate is the “width” of the field at each grid point; the complete grid therefore represents the “width” variation of the field as a function of  $x$  and  $y$ . The important quantity to model is the ratio of cross-sectional areas between two adjacent axial stations. This means that the  $z$ -coordinate may be scaled with no effect on the computed flowfield, but a simple translation of the  $z$ -coordinates will change the computed flowfield, because the cross-sectional area ratio will be different. As with two-dimensional calculations, the value of the  $z$ -coordinate will affect flux-related post-processing calculations.

#### 3.1.4 Axisymmetric Cases

Axisymmetric configurations may be modeled using a two-dimensional grid generated at an arbitrary circumferential location on the geometry — e.g., the top centerline. Note that the grid should be generated on only one “side” of the configuration. Once again, the  $z$ -coordinate of the grid should be 1.0. The final step in using WIND’s axisymmetric mode is the specification of the symmetry axis location and the circumferential sweep angle in the input data file. The circumferential sweep angle is the angle of the “pie shape” swept out by the grid about the symmetry axis. Although the value of the sweep angle will not affect the computed flowfield, it will affect flux-related post-processing calculations.

*Keywords:* [AXISYMMETRIC](#)

## 3.2 Euler and Navier-Stokes Equations

WIND may be used to solve the Euler equations or the Reynolds-averaged form of the Navier-Stokes equations (Bush, 1988). All heat transfer and stress tensor terms are retained, and the equations are modeled in full conservation form. The effects of turbulence may be modeled using a variety of algebraic, one-equation, and two-equation turbulence models. Modification of the effective heat transport coefficient due to turbulence is linked to the momentum diffusion coefficient by a turbulent Prandtl number, which is assumed to be constant.

The fluid may be treated as a thermally and calorically perfect gas, a thermally perfect gas, equilibrium air, or a mixture undergoing a finite rate chemical reaction. For an ideal gas, conventional values are given to the gas constant ( $R$ ) and the ratio of specific heats ( $\gamma$ ), or they may be specified. Effects of gravity (i.e., stratification) and rotation may also be included.

The equation set(s) to be solved must be specified in the input data file.

*Keywords:* [TURBULENCE](#), [GRAVITY](#), [ROTATE](#)

### 3.2.1 Freestream Conditions

Freestream flowfield conditions — Mach number, pressure, temperature, angle of attack, and angle of sideslip — must be specified in WIND's input data file. The Mach number must be greater than 0, and pressure and temperature may be specified as static or total values. These conditions are used to initialize the flowfield at the start of a run. For external flow problems, they are also applied at all inflow, outflow, and freestream boundaries during the course of a flow solution. For this reason, the outermost grid boundary should be far enough away from the body such that the freestream assumption is valid at the boundary.

*Keywords:* [FREESTREAM](#)

### 3.2.2 Reynolds Number Considerations

WIND does not allow the direct specification of a Reynolds number; freestream conditions must be specified that are consistent with the desired value. Currently, WIND uses a reference length which it computes based on the unit system of the grid file. The code prints out the reference length (in inches) that it uses, so that you may compare the Reynolds number from the code with the desired value. The Reynolds number in WIND's output is actually the Reynolds number "per reference length."

Here are a couple of examples.

*Case I: Grid is same size as model*

Suppose we have the grid for a wind tunnel model, and we want to run it at  $M_\infty = 0.7$  and a Reynolds number of 12.1 million (based on the model length). We arbitrarily choose a total temperature of 520 °R (static temperature of 473.6 °R). Knowing the temperature, we can calculate the speed of sound ( $a = \sqrt{\gamma RT}$ ) and viscosity (from Sutherland's law). Using the Mach number, we

can calculate the freestream velocity. Thus, we have

$$\begin{aligned} M_\infty &= 0.7 \\ T_\infty &= 473.6 \text{ }^\circ\text{R} \\ U_\infty &= 746.67 \text{ ft/sec} \\ \mu_\infty &= 3.474 \times 10^{-7} \text{ lb}_f\text{-sec/ft}^2 \\ Re_L &= 12,100,000 \end{aligned}$$

We now have the Mach number and temperature for the WIND input data file, but we still need to calculate the freestream pressure. Using the definition of the Reynolds number (and the ideal gas law),

$$Re_L = \frac{\rho U L}{\mu} = \frac{P U L}{R T \mu}$$

or

$$P = \frac{R T \mu Re}{U L} = \frac{R T \mu Re_L}{U}$$

Note that  $Re_L$  is the Reynolds number per unit length of the physical model. For our example, if the model length is 10 inches,

$$\begin{aligned} Re_L &= 12,100,000/(10/12 \text{ ft}) \\ &= 14,520,000(1/\text{ft}) \end{aligned}$$

We can now calculate  $P$ :

$$\begin{aligned} P &= \frac{R T \mu Re_L}{U} = \frac{(1716)(473.6)(3.474 \times 10^{-7})(14.52 \times 10^6)}{746.67} \\ &= 5490.31 \text{ lb}_f/\text{ft}^2 = 38.13 \text{ psi} \end{aligned}$$

We would now like to check our input. If we run WIND with the Mach, temperature and pressure specified above, the code will print a Reynolds number and a reference length near the top of the list output file. *To obtain the desired Reynolds number, divide WIND's value of the Reynolds number by the output reference length and multiply by the model body length. This number may be compared with the desired model Reynolds number.*

#### Case II: Grid is scaled from model size

Let us now assume that we want to run the previous grid at flight conditions, but we want to keep our same old 10-inch grid. We simply need to multiply the pressure by a scale factor. The equation now becomes:

$$P = \frac{R T \mu Re_L S}{U}$$

where

$$S = \frac{\text{Full Model Size}}{\text{Grid Model Size}}$$

For example, if we want to run a 100-inch wing using our 10-inch grid,  $S = 10$ . If we want to run a flight Reynolds number of 26 million, we calculate  $P$  as:

$$\begin{aligned}
 P &= \frac{(1716.5)(473.6)(3.47 \times 10^{-7})(26 \times 10^6)(10)}{746.67} \\
 &= 983113.0 \text{ lb}_f/\text{ft}^2 = 682.72 \text{ psi}
 \end{aligned}$$

### 3.2.3 Mass Flow in Two-Dimensional Calculations

One of the options available in WIND is the specification of mass flow boundary conditions for subsonic duct analyses. Actual or corrected mass flow may be specified at duct exits, as may back pressure. Regardless of the exit condition specified, WIND translates it into a uniform static pressure at the exit (which may or may not remain constant).

Within WIND and many post-processors, routines exist which integrate mass flow at desired computational planes. For 3D cases, the desired mass flow may be compared directly with the output from the integration routines:

$$\dot{m} = \int \rho u dA$$

However, for 2D calculations, the comparison is not so straightforward. There are three cases to consider.

#### Case I: 2D, Unit Depth

The first case involves running WIND on a truly two-dimensional grid of *unit depth* ( $z$ -coordinate is 1.0 everywhere). In this case, the input mass flow should be *per unit depth*. For example, let's say we want to run a 2D, unit depth model of a duct with a square exit. (If the exit were not square, this model would probably not be very good.) We would like a corrected mass flow of 500 lb<sub>m</sub>/sec, and our actual model exit depth is 10 inches. If the grid input units are inches, we should ask for a mass flow of 50 lb<sub>m</sub>/sec. If the grid input units are *not* inches, simply divide the actual mass flow by the  $z$ -coordinate value *in inches*.

#### Case II: 2D, Variable Width

When the  $z$ -coordinate is the width of the 2D grid, WIND adds in the area variation as a source to the 2D equations, making the analysis quasi-three-dimensional. In this case, the actual 3D mass flow should be specified in the input file. The integrated exit area will be (approximately, see [Section 3.2.4](#) the real duct exit area, if the width has been specified correctly.

#### Case III: 2D, Axisymmetric

Axisymmetric runs require specification of the symmetry axis location and the circumferential angle subtended by the 2D grid. (This angle has no influence on the solution, but it does determine the area perpendicular to the grid.) The only reason this angle is an input parameter is so that you will know what the streamwise area is. In this case, the real exit geometry is circular, with a corresponding mass flow. The ratio of the input mass flow to the actual mass flow should equal the ratio of the input circumferential angle to 360. For example, if we are modeling a circular duct with a mass flow of 200 lb<sub>m</sub>/sec using an axisymmetric model in WIND, and if we specify a circumferential angle of 36 (1/10 of 360), we should specify a mass flow of 20 lb<sub>m</sub>/sec (1/10 of 200 lb<sub>m</sub>/sec).



#### 3.2.4 Mass Flow and Grid Areas

When dealing with subsonic duct analyses, you should be aware that the duct area as represented by the grid may be slightly different from the real area of the geometry being modeled, especially for ducts modeled with quasi-polar grids.

The duct area represented by the computational grid is often smaller than the real duct area, which, when running near critical mass flow, may prematurely choke the flow in the CFD solution. If the duct is circular and is modeled with a quasi-polar grid, the area error may be estimated.

Suppose we are modeling a circular duct with a quasi-polar grid using  $k_{max} = 33$  circumferential points, each of which lie on the perimeter of the real duct at some streamwise station. If the circumferential points are evenly distributed, we may describe this topology as a circle which circumscribes a regular polygon of  $k_{max} - 1 = 32$  sides. For a circle of radius  $R$  circumscribing a regular polygon of  $n$  sides, the area of the polygon is

$$A_p = \frac{1}{2} n R^2 \sin \frac{360}{n}$$

which means that the “grid area” is

$$A_g = \frac{1}{2} (k_{max} - 1) R^2 \sin \frac{360}{k_{max} - 1}$$

For our example, with  $k_{max} = 33$ , the grid area is 0.7% lower than the actual area.

There is no need to worry about this difference for most cases, but you should be aware of its possible effects.

#### 3.2.5 Heat Transfer

At solid walls, WIND uses an adiabatic heat transfer boundary condition by default. A constant wall temperature may also be specified in the input data file. Through the use of the [TTSPEC](#) keyword, point-by-point wall temperature distributions may also be specified on boundary surfaces. An auxiliary code, [tmptm](#), is used to create the wall temperature distribution, and write it into the common flow (.cfl) file.

The thermal conductivity is determined using a constant Prandtl number.

*Keywords:* [WALL TEMPERATURE](#), [TTSPEC](#)

#### 3.2.6 Viscosity

By default, WIND uses Sutherland’s law to define laminar viscosity as a function of temperature. Keye’s formula may be used in addition to Sutherland’s law, and Wilke’s law may be used to compute the laminar viscosity for multi-species flows.

*Keywords:* [VISCOSITY](#)

### 3.3 Turbulence Models

For turbulent calculations, Reynolds averaging assumptions are used to define a turbulent (eddy) viscosity, which is added to the laminar viscosity in the flow calculations. All the turbulence models

available in WIND are coupled to the Navier-Stokes equations only through the turbulent viscosity. Several algebraic, one-equation, and two-equation turbulence models are available.

Note that a turbulence model (or inviscid or laminar flow) must be specified in the input data file. WIND will stop if you do not.

### 3.3.1 Algebraic Models

The algebraic turbulence models available in WIND are the Cebeci-Smith model, the Baldwin-Lomax model ([Baldwin and Lomax, 1978](#)), and the P. D. Thomas model ([Thomas, 1979](#)), which adds a shear layer model to the Baldwin-Lomax model. After each iteration of the flow solver, these models compute the turbulent viscosity based on current flowfield quantities. Note that, because of their dependence on maxima and minima of flowfield variables, these models produce discontinuous turbulent viscosity distributions in the computed flowfield and require special numerical treatment at the juncture of two walls and on computational  $i$ -boundaries (the latter for historical reasons). The Baldwin-Lomax model is the most widely used algebraic turbulence model in WIND.

*Keywords:* [TURBULENCE](#)

### 3.3.2 One-Equation Models

Because of their efficiency and ability to produce continuous turbulent viscosity distributions, the one-equation turbulence models in WIND are the models of choice for many engineering applications. The one-equation models available in WIND are the Baldwin-Barth ([Baldwin and Barth, 1990](#)) and Spalart-Allmaras ([Spalart and Allmaras, 1992](#)) models.

For three-dimensional unsteady flows, the Spalart Detached Eddy Simulation (DES) model may be used. ([Spalart, Jou, Strelets, and Allmaras, 1997](#); [Shur, Spalart, Strelets, and Travin, 1999](#)) This reduces to the standard Spalart-Allmaras model near viscous walls, where the grid is refined and has a large aspect ratio, but acts like a Large Eddy Simulation (LES) model away from the boundary, where the grid is coarser and has an aspect ratio of order one. It is intended to improve the results for unsteady and massively separated flows.

*Keywords:* [TURBULENCE](#)

### 3.3.3 Two-Equation Models

Two two-equation turbulence models are currently available in the WIND code — the Menter Shear Stress Transport (SST) model ([Menter, 1993](#); [Mani, Ladd, Cain, and Bush, 1997](#)) and the low-Reynolds-number Chien  $k$ - $\epsilon$  model ([Chien, 1982](#)).

The Menter Shear Stress Transport (SST) model is a blend of  $k$ - $\epsilon$  and  $k$ - $\omega$ . The equations are cast in  $k$ - $\omega$  form. Near the boundary the  $k$ - $\omega$  model is used, and the  $k$ - $\epsilon$  model is used away from the wall and in shear layers. Freestream values of  $k$  and  $\omega$  may be specified. Each equation is solved individually and an iterative method has been used on the left-hand side to reduce the factorization error. The model is robust, and may be more accurate in adverse pressure gradients than some of the other models in WIND.

For unsteady flows, a combined SST and Large Eddy Simulation (LES) model may be used. The combined model reduces to the standard SST model in high mean shear regions (e.g., near viscous walls), where the grid is refined and has a large aspect ratio unsuitable for LES models. As the grid is traversed away from high mean shear regions, it typically becomes coarser and more isotropic, and

the combined model smoothly transitions to an LES model. The intent is to improve predictions of complex flows in a real-world engineering environment, by allowing the use of LES methods with grids typical of those used with traditional Reynolds Averaged Navier Stokes models.

The Chien  $k$ - $\epsilon$  model is also available in WIND. Several options may be specified with this model to control the initialization procedure, enhance its stability, and improve its accuracy in adverse pressure gradients and at high Mach numbers.

*Keywords:* **TURBULENCE**

#### 3.3.4 Transition Specification

Through the use of the **TTSPEC** keyword, point-by-point transition data may be specified on viscous walls. The data represent the percentage of turbulent viscosity to be added to the laminar viscosity at each grid point. An auxiliary code, *tmptrn*, may be used to create the transition data, and write it into the common flow (.cfl) file.

*Keywords:* **TTSPEC**

### 3.4 Gas Models

A variety of gas models are available in WIND to complete the equation set. The fluid may be treated as a thermally and calorically perfect gas, a thermally perfect gas (frozen chemistry), equilibrium air, or a mixture undergoing a finite rate chemical reaction. Several different chemistry packages are available as files containing thermodynamic data, finite rate coefficients, and transport property data.

*Keywords:* **CHEMISTRY**

### 3.5 Flowfield Initialization

By default, WIND initializes the computational flowfield by setting the flow properties at each grid point equal to those specified with the **FREESTREAM** keyword in the input data file. The same initial conditions are applied at all points in the computational domain, including solid walls, zone boundaries, and freestream boundaries. Several options in WIND lead to non-default initializations, including user-specified inflow conditions, boundary layer initialization, and reinitialization of portions of the flowfield on restart.

#### 3.5.1 User-Specified Initialization

WIND's **ARBITRARY INFLOW** keyword was designed to specify flow conditions at **arbitrary inflow** boundaries.<sup>13</sup> However, it may also be used in a variety of ways to specify initial conditions in selected portions of the computational domain that are different from the freestream values. This capability may be useful, for example, when modeling a jet emanating from a solid wall. Zones downstream of the jet exit may have difficulty converging to the proper solution from conditions much different than the jet exit conditions.

<sup>13</sup>Recall that the type of boundary, such as **arbitrary inflow**, is specified using GMAN, and stored in the common grid (.cgd) file, not in the input data file.

To use the **ARBITRARY INFLOW** keyword as an initialization tool, specify the appropriate parameters and values for a zone as described below, and run WIND from scratch (i.e., without an existing flow (.cfl) file).<sup>14</sup>

There are three ways that the **ARBITRARY INFLOW** keyword may be used to modify the default initial conditions.

- If uniform inflow is specified for a particular zone using the **UNIFORM** parameter, the flow *throughout that zone* will be initialized to the conditions specified.
- The **IJK\_RANGE** parameter may be used to set flow conditions over an arbitrary range of  $i$ ,  $j$ , and  $k$  indices in any computational region. Up to 500 **IJK\_RANGE** parameters are allowed. The default initial conditions will be used at points outside the specified range(s).
- The **USERSPEC** parameter may be used to specify a 1-D profile normal to the surface, translated through some buttline range, below the vehicle. Note that this option only applies to points in the  $i = 1$  computational plane. The default initial conditions will be used at the remaining points.

*Keywords:* **ARBITRARY INFLOW**

### 3.5.2 Boundary Layer Initialization

To provide a better approximation to near-wall flowfields, WIND provides a couple of boundary layer initialization options that may help speed the convergence of viscous flows near solid walls.

First, by using multiple **IJK\_RANGE** parameters with the **ARBITRARY INFLOW** keyword, and setting the range to include not only the inflow boundary but also locations downstream, one can set initial boundary layer profiles all along the viscous walls. Note that using this capability may add many lines to the input data (.dat) file. The **INCLUDE** keyword may be useful in keeping the main .dat file to a manageable size.

Another option is to use the **BL\_INIT** keyword to specify a starting location and thickness for a laminar or turbulent boundary layer. However, this option may only be used on computational  $j$ - or  $k$ -boundaries, and only on one boundary in each zone.

*Keywords:* **ARBITRARY INFLOW**, **BL\_INIT**

### 3.5.3 Reinitialization

In the event that portions of the computed flowfield become “polluted” with unrealistic flowfield data, due to numerical instabilities or other causes, you may wish to reinitialize portions of the flow. WIND’s reinitialization option enables you to reset the flow conditions in specified zones. The reinitialization will be done as if WIND were being run from scratch. The conditions at all or specified grid points in the specified zones will be reinitialized to either freestream values, or to values specified using the **ARBITRARY INFLOW** keyword and/or the **BL\_INIT** keyword, as described above.

*Keywords:* **REINITIALIZE**, **ARBITRARY INFLOW**, **BL\_INIT**

---

<sup>14</sup>Note that since the **ARBITRARY INFLOW** keyword is also used to specify boundary conditions at **arbitrary inflow** boundaries, a conflict arises if (for some reason) the desired inflow properties are different from those being set as initial conditions. In this case, the **ARBITRARY INFLOW** keyword can still be used to set initial conditions by setting the number of cycles to be run to zero. Then, after changing the values specified with the **ARBITRARY INFLOW** keyword to the desired inflow values, simply restart using the initialized flowfield in the newly-created .cfl file.

## 4 Numerical Modeling

WIND utilizes a number of different numerical methods in the solution of a selected set of field equations. By default, the code solves the Euler and Navier-Stokes equations using Roe’s flux-difference splitting scheme for the explicit (right-hand side) terms, and either a diagonalized or block matrix solver for the approximately-factored implicit (left-hand side) terms.

### 4.1 Iterations and Cycles

The usual method of obtaining a solution with WIND involves the initialization of the entire flowfield and the successive iteration of the flow solver, with appropriate explicit boundary conditions, to a steady-state solution. Most boundary conditions are updated after each iteration of the flow solver; however, because of its expense, zone coupling takes place less frequently. Zone coupling is the process whereby WIND exchanges flowfield information between zones to maintain solution continuity at zonal interfaces. Updates of the flow file with the most current data also take place less frequently.

To facilitate flow file (see [Section 7.3](#)) updates and propagation of flowfield information to neighboring zones, WIND defines a “cycle” as a specific number of iterations performed in each of the zones, with zone coupling to effect information transfer. By default, WIND performs five iterations in each zone before moving on to the next zone. You must specify the number of solution cycles to be computed, and you may optionally specify the number of iterations per cycle to be calculated in each zone (or all zones simultaneously).

For single-zone problems, you may wish to specify a large number of iterations per cycle, because there is no need for zone coupling operations; however, note that the flow file will be updated less frequently in this case. For multi-zone problems, a large number of iterations per cycle is not recommended, because it will greatly hinder the transfer of flowfield information between zones and, therefore, convergence.

#### 4.1.1 “Turning Off” Zones

To “turn off” a zone in a multi-zone calculation, simply specify 0 or fewer iterations per cycle for that zone; values of  $-1$  and  $-2$  are used to indicate how zone coupling is to be handled for zones adjacent to the “dead” zone.

*Keywords:* [CYCLES](#), [ITERATIONS](#)

### 4.2 Grid Considerations

WIND provides capabilities which allow you to make the most efficient use of the computational grid for a given configuration. Both of the options described in this section reduce the accuracy of the computations in one or more grid directions and should therefore not be used haphazardly.

#### 4.2.1 Grid Sequencing

Grid sequencing is a capability whereby WIND “removes” grid points from selected portions of the domain, resulting in a “coarse grid” solution which is obtained in a fraction of the time it would have taken for a “full grid” solution. For example, if you select one level of sequencing in all three

directions in a particular zone, only every other grid point in each of the  $i$ -,  $j$ -, and  $k$ -directions will be used in the calculations, resulting in a decrease in CPU time of approximately 80–90%. (The speed-up is not a linear function of the number of sequencing levels.) Selecting two levels of sequencing would repeat the process, causing only every fourth grid point to be used. At the end of each run, the solution is interpolated back onto the entire grid to aid in restarting the solution and to provide a continuous field for post-processing.

Grid sequencing is very useful for converging gross flowfield properties before obtaining a final solution on the full grid. Note, however, that a sequenced solution may seem incorrect when post-processed using the entire computational grid, because of the interpolation process used at the end of each run. In addition, when restarting a full-grid solution from a sequenced solution, reducing the CFL number (time step) by 50% is highly recommended, in order to reduce the risk of solution instabilities which arise during the transition.

*Keywords:* [CFL#](#), [SEQUENCE](#)

### 4.2.2 Thin-Shear-Layer Calculations

Another time-saving option available in WIND is the ability to solve the thin-shear-layer Navier-Stokes equations, instead of the full equations. By requesting that WIND remove the viscous terms from the equations in one or more computational directions, a CPU savings of up to 30% may be realized. This capability requires careful specification, because viscous terms must be retained along grid lines which are perpendicular to boundary layers, free shear layers, and other highly viscous flow features.

Because some solution instabilities have been linked to the use of the thin-shear-layer option, it is recommended that you disable this option near the completion (convergence) of your solution, creating a fully viscous flowfield model.

*Keywords:* [TSL](#)

## 4.3 Explicit Operator

The first-derivative (non-viscous) terms on the right-hand side of WIND's equation set are calculated by the code's explicit operator. You may choose from a wide variety of explicit operators, independently of the implicit (left-hand-side) operator. An explicit operator is selected by its type and accuracy. Types include a central difference operator, the upwind schemes of Coakley, Roe, and Van Leer, and a modified version of the Roe scheme for stretched grids. Depending on the type of scheme used, the accuracy may be specified as anywhere from first to fifth order. The default explicit operator is Roe's second-order flux-difference splitting algorithm, modified for stretched grids.

*Keywords:* [RHS](#)

### 4.3.1 Explicit Smoothing

Certain types of explicit operators — for example, the central difference operator — may require the addition of numerical smoothing in order to dampen instabilities which are a natural part of the scheme. Smoothing must also be added explicitly when utilizing WIND's convergence acceleration capability. Values for second- and fourth-order smoothing coefficients and a smoothing limiter coefficient may be specified in the input data file.

*Keywords:* [ACCELERATE](#), [SMOOTHING](#)

### 4.3.2 Explicit Boundary Damping

In flowfields dominated by acoustic phenomena, unwanted wave reflections may sometimes occur at freestream computational boundaries. To eliminate these unphysical reflections, WIND’s boundary damping capability may be employed to apply explicit smoothing near the freestream boundaries, with the effect of absorbing the reflected waves.

*Keywords:* [BOUNDARY-DAMP](#)

### 4.3.3 Total-Variation-Diminishing (TVD) Operator

Certain types of explicit operators are equipped with a Total-Variation-Diminishing (TVD) operator, which limits local maxima and minima in flux quantities to prevent non-physical instabilities from arising during the solution. The default Roe scheme, for example, requires interpolation and/or extrapolation of flux quantities, which may produce minima or maxima that are outside the range of the original data. TVD operators attempt to limit these peaks to acceptable values based on the data used in the interpolation and extrapolation. TVD may be specified on a zonal basis in the WIND input data file, and the severity of its limiting may be controlled through an input parameter.

Because only single data points are used in the first-order schemes, TVD, which limits interpolations and extrapolations, has no effect on such cases. It will only have an effect for second- and higher-order calculations. When attempting to model precise phenomena, such as acoustic waves, TVD should *not* be used, because it may produce undesirable damping or smearing of flowfield features. However, in most cases, TVD provides significant protection against numerical instabilities with less than a 10% increase in CPU time.

*Keywords:* [TVD](#), [BOUNDARY TVD](#)

## 4.4 Implicit Operator

The left-hand side of WIND’s equation set is computed in the code’s implicit operator, which may be set on a zonal basis. A variety of implicit operators are available, including point Jacobi, Gauss-Seidel, and MacCormack modified approximate factorization.

Also available are options to turn off the implicit operator completely for an entirely explicit calculation, or to select scalar, full block, or parabolized Navier-Stokes implicit operators. With these options, a different implicit operator may be specified for each computational direction.

By default, WIND uses the full block operator in viscous directions, and the scalar (diagonalized) matrix solver in inviscid directions.

*Keywords:* [IMPLICIT](#)

### 4.4.1 Implicit Boundaries

In addition to the normal explicit boundary conditions, implicit boundary conditions may be used on “wall” boundaries. These improve stability when the CFL number is above about 1.3.

*Keywords:* [IMPLICIT BOUNDARY](#)

## 4.5 Time Step

### 4.5.1 CFL Number

The time step can be one of the most important parameters in a WIND solution. It may mean the difference between a successful run and a code bomb. In WIND, the time step is represented by the CFL (Courant-Friedrichs-Lewy) convergence criterion, which is a non-dimensional time step based on the actual time step, the local grid spacing, and the local characteristic velocity ( $u + c$ ).

For stability reasons, WIND's default CFL number is set to 1.3, which is sufficient for a wide variety of problems. For inviscid solutions, the CFL may be raised slightly, while for highly complex flows (due to complex geometry or gas models), it may need to be lowered significantly. Often, in the beginning of a solution, or during the transition between a sequenced and fine grid solution, the CFL number should be set to 0.5 while transients settle out of the flowfield.

Note: If you must consistently set the CFL number to 0.1 or lower to maintain stability in the solution, you should attempt to locate other possible causes of numerical instability, such as grid anomalies or grid/flow interactions. Such low CFL numbers translate into small time steps and unacceptable convergence rates.

When a CFL number above 1.3 is desired, the implicit boundary conditions should be used.

By default, WIND defines the CFL number based on the minimum time step associated with the eigenvalues of each coordinate direction individually. That is:

$$dt = \min \left( \frac{CFL_x}{\lambda_x}, \frac{CFL_y}{\lambda_y}, \frac{CFL_z}{\lambda_z} \right)$$

Using [TEST 105](#), the user may define the time step more conventionally as:

$$dt = \frac{CFL}{\lambda_x + \lambda_y + \lambda_z}$$

which provides a somewhat smaller effective time step.

For most calculations, the time step is computed based on a constant CFL throughout the field, which means that, for non-uniform grids, the solution advances at different rates in different parts of the grid. Because the CFL number is directly proportional to the physical time step and inversely proportional to the local grid spacing, the physical time step must be small where the grid spacing is small, in order to maintain a constant CFL number. For example, in boundary layers and shear layers, where the grid is often closely packed for better resolution, the solution advances more slowly than in other parts of the flowfield. In fact, most of the iterations in a viscous solution are spent converging viscous regions. In practice, the major flow features (e.g., shocks and other major pressure gradients) will develop quickly, while detailed flow phenomena will require a larger number of iterations.

WIND therefore uses by default a “cross-flow” CFL number which multiplies each coordinate direction's eigenvalues by a factor before determining the time step associated with that spacing and flow conditions. This factor varies from 1.0 when the flow is along the coordinate direction, to a user-specified value when the flow is perpendicular to the coordinate direction. The far-field CFL number is generally the value specified in the input data file with the [CFL#](#) keyword, but near the wall the code effectively uses a larger value. This should increase convergence of the boundary layer.

The cross-flow CFL factor is defined using the [CROSSFLOW](#) keyword. Values greater than one effectively increase the time step in the boundary layer (where the flow is parallel to the wall, but



the minimum spacing is normal to the wall) by the specified factor. The default is 2.0 for three-dimensional flow, and 1.0 (i.e., no time step increase in the boundary layer) for two-dimensional flow and axisymmetric flow.

*Keywords:* CFL#, CROSSFLOW

#### 4.5.2 Runge-Kutta Time Step

Several Runge-Kutta time step formulations are available for use in WIND. A traditional four-stage Jameson-type scheme or a three-stage, minimal-storage scheme may be selected in the input data file. You may also specify algorithm coefficients for each of the stages in the different schemes. Runge-Kutta schemes attempt to provide a more accurate approximation to the solution at the next time level by advancing through several stages, or sub-iterations, to get there. In WIND, each sub-iteration may be one pass through the explicit operator with or without a pass through the implicit operator. (The original Runge-Kutta scheme was formulated using only an explicit operator.) Depending on the coefficients used, Runge-Kutta time step algorithms are often higher order in time than the default Euler time stepping algorithm. This makes them ideally suited for time-accurate calculations (see below).

*Keywords:* STAGES

#### 4.5.3 Global Newton Iteration

A Global Newton iteration technique is available in WIND, and has proven to provide time-accurate solutions with large CFL numbers for unsteady flows with large time scales (Tramel and Nichols, 1997). The Global Newton technique stabilizes the solution and improves time accuracy by placing the entire unsteady transport equations on the right-hand side of the matrix solver, and iterating within a time step over all the zones. Thus, the interface boundaries are brought up to the new time level, along with the interior flow field, resulting in an essentially implicit treatment of the boundaries. The Global Newton algorithm has also been shown to improve steady-state convergence rates (Nichols and Tramel, 1997).

*Keywords:* NEWTON

#### 4.5.4 Time-Accurate Solutions

The objective of time-accurate CFD calculations is the modeling of unsteady flow phenomena, such as vortex pairing or bluff body wakes. In these calculations, the physical time step must be the same throughout the entire flowfield, regardless of the local grid distribution. Because the CFL number has a limit which approximates the upper limit of numerical stability, the time step used in the calculation must be obtained from the maximum CFL number and the smallest grid spacing in the field. Thus, the stability of the calculations in regions of dense grid packing determines the time step for the entire rest of the field. For single-zone time-accurate calculations, you may request that WIND calculate the minimum time step in the zone based on the specified CFL number. However, for multi-zone time-accurate calculations, you must calculate the time step yourself, based on the desired CFL number and the minimum grid spacing, and specify the physical time step in the input data file.

*Keywords:* CFL#

## 4.6 Convergence Acceleration

The solution transients that occur in iterating from the freestream-initialized flowfield often cause the most severe stability problems in WIND. As the solution approaches convergence on a sequenced grid, most of these transients will have settled out of the field, with the change to fine grid being the only remaining “jolt” to the solution. Numerical stability may be forced through the addition of artificial numerical dissipation to the equations, but such dissipation may lead to non-physical results in the steady state solution.

WIND's convergence acceleration capability was developed to make use of artificial dissipation to eliminate stability problems, while raising the initial solution time step to get past the transients more quickly. In the input data file, you may specify the initial amount of artificial dissipation (smoothing) to be added to the equations, the starting and ending values of the CFL number, the iteration numbers between which the smoothing should be applied and CFL number altered, and the zone(s) in which you would like to enable this capability. Recommended values for the smoothing coefficients are given later in this manual, under the **ACCELERATE** keyword description (see p. 77).

At the beginning of the solution, using this capability, a relatively high CFL number will be accompanied by significant numerical smoothing. As the iteration count approaches the specified ending value, the CFL number will be gradually adjusted toward its final value as the amount of smoothing is steadily decreased.

*Keywords:* **ACCELERATE**, **SMOOTHING**, **TEST 49 2**

## 5 Boundary Conditions

The boundary conditions are perhaps the most important factor in influencing the accuracy of the flow computation. The manner in which the boundary conditions are imposed also influences the convergence properties of the solution. WIND uses a cell-vertex discretization, which results in solution points located on the boundaries of the zones which comprise the flow domain. During the computation, WIND computes the boundary values for the conservative variables, the species (if present), and turbulence variables (if present).

Proper specification of the flow boundary conditions is aided by a basic understanding of characteristic theory of the incoming and outgoing waves normal to the boundary. The boundary normal is considered positive when it points into the flow domain. The wave speeds (eigenvalues) have convective and acoustic components. A positive wave speed indicates a wave entering the flow domain, and so, a physical boundary condition must be specified and some auxiliary information must be supplied to impose the boundary condition. A negative eigenvalue indicates a wave leaving the flow domain, and so, a numerical boundary condition can be specified using flow data within the domain. The waves moving tangential to the boundary are neglected in the boundary condition treatment.

### 5.1 Explicit and Implicit Boundary Conditions

WIND imposes the boundary conditions explicitly after the interior solution has been computed for each zone after each iteration, by default. An exception is the zone interface boundaries, which are updated after each cycle. Another exception is the mass flow boundary condition, which is updated after a specified number of iterations (the default is five iterations) in order to reduce computational effort. Errors due to explicit boundary conditions are reduced through the use of the multi-stage or iterative time integration methods.

The use of implicit boundary conditions with an implicit solver is known to improve the stability of the method and lead to faster convergence at the expense of greater computational effort and complexity. Implicit boundary conditions are available in WIND in a limited manner, primarily for wall boundary conditions. The `IMPLICIT BOUNDARY` keyword is used to turn on implicit boundary conditions.

*Keywords:* `IMPLICIT BOUNDARY`

### 5.2 Boundary Condition Types

The Grid MANipulation (GMAN) program is used to associate boundary condition types with the solution points on the boundaries of the zones. These boundaries represent the geometry model, fluid boundaries, grid topological boundaries, and couplings between zones. The type of boundary conditions available in GMAN, in the order displayed by GMAN, include<sup>15</sup>:

- `undefined`
- `reflection`
- `freestream`
- `viscous wall`
- `arbitrary inflow`

---

<sup>15</sup>In this User's Guide, GMAN boundary condition types are indicated by lower-case words in a fixed-pitch font, like `this`; WIND keywords are displayed in an upper-case fixed-pitch font, `LIKE THIS`.

- [outflow](#)
- [inviscid wall](#)
- [self-closing](#)
- [singular axis](#)
- [coupled](#)
- [bleed](#)
- [pinwheel axis](#)
- [frozen](#)
- [chimera](#)

It is possible to group some of these boundary condition types in a logical manner for detailed discussions, which are presented below. Some boundary conditions require further information, which is provided through keywords in the input data file, beyond being flagged in GMAN.

### 5.3 Wall Boundary Conditions

The [inviscid wall](#), [viscous wall](#), and [bleed](#) boundary condition types are all wall boundary conditions which simulate interaction of the flow with a real or imaginary solid surface.

#### 5.3.1 Inviscid Wall

The [inviscid wall](#) boundary condition imposes flow tangency at the zone boundary (wall surface) while maintaining the same total velocity as the point adjacent to the boundary. One numerical boundary condition is imposed by computing the pressure at the boundary through an interpolation of interior pressures. A zero-order extrapolation is robust; however, the pressures may not be smoothly varying at the boundary. A first-order extrapolation works well for flows without discontinuity, and for flows in which the pressure does not vary greatly normal to the boundary. The extrapolations across a discontinuity may result in nonphysical pressures. One can use the [viscous wall](#) boundary condition along with the [TURBULENCE INVISCID](#) keyword in the input data file. This is useful if one wants to start a computation with the boundary as inviscid and then later turn on the viscous boundary conditions. In WIND, it is also possible to specify through [TEST 138](#) that the normal pressure gradient at the wall be calculated rather than simply assuming it to be zero. Also, a first-order extrapolation which accounts for grid spacing can be used through the use of [TEST 141](#).

#### 5.3.2 Viscous Wall

The [viscous wall](#) boundary condition imposes a no-slip condition of the flow, a zero pressure gradient, and the appropriate heat transfer condition (adiabatic or constant temperature) at the zone boundary (wall surface). The no-slip condition can involve a non-zero velocity if the wall is moving. (See the [MOVING WALL](#) and [ROLL](#) keywords; however, for moving walls, the boundary condition type should be set to [bleed](#).) To minimize transients at the start of a WIND calculation, the velocity at no-slip boundaries is actually reduced from its initial value to the no-slip condition over a number of iterations. The number of iterations may be specified using the [WALL SLIP](#) keyword.

The choice of the heat transfer condition is determined through the use of the [WALL TEMPERATURE](#) keyword in the input data file. The default is an adiabatic wall (zero temperature gradient). The

temperature for the constant temperature condition is specified through the `WALL TEMPERATURE` keyword. The `TTSPEC` keyword is available for specifying a point-by-point distribution of surface temperatures.

Wall function boundary conditions may be used at viscous walls, using the White-Christoph law of the wall, through the `WALL FUNCTION` keyword. This feature is currently available for single-species flows only.

In WIND, it is also possible to specify through `TEST 138` that the normal pressure gradient at the wall be calculated rather than simply assuming it to be zero. Also, a first-order extrapolation which accounts for grid spacing can be used through the use of `TEST 141`. Viscous flow is computed when the `TURBULENCE` keyword is used.

*Keywords:* `MOVING WALL`, `TTSPEC`, `TURBULENCE`, `WALL FUNCTION`, `WALL SLIP`, `WALL TEMPERATURE`

### 5.3.3 Bleed

The `bleed` boundary condition allows mass to flow through a porous, viscous wall. Bleed is mass flow out of the flow domain, while blowing is mass flow into the flow domain. Bleed and blowing systems are often an integral part of aeropropulsion configuration design, helping to control such flow phenomena as boundary layer growth and mixing. WIND's bleed / blowing boundary condition was designed to provide a means to model these systems with CFD. Specification of the `bleed` boundary condition in GMAN, which involves the identification of a particular bleed region number, triggers the calculation of the area of the bleed region specified, which is then stored in the grid file. WIND uses this area and the bleed or blowing conditions specified in the input data file to compute a normal velocity on the model surface. Bleed may be specified as a mass flow, or as a porous surface with a discharge coefficient and back pressure. Blowing may be specified through mass flow or plenum conditions.

*Keywords:* `BLEED`, `BLOW`

**Moving Wall.** The `moving wall` boundary condition enables a tangential velocity to be applied at no-slip walls in order to model rotating hubs or other components in the flow. The boundary solution points for the moving wall should be identified as `bleed` regions in GMAN. The translating or spinning motion of the wall is specified through the `MOVING WALL` keyword in the input data file. The `ROLL` keyword allows a rolling motion to be imposed on the grid.

*Keywords:* `MOVING WALL`, `ROLL`

## 5.4 Flow Interface Boundary Conditions

The `freestream`, `arbitrary inflow`, and `outflow` boundary condition types form a group involving the simulation of the interaction of the flow with other flow conditions at the domain boundaries.

### 5.4.1 Freestream

The `freestream` boundary condition is intended for use at freestream outer boundaries, and may also be useful at inflow boundaries. This boundary condition uses one-dimensional characteristic theory to set boundary flowfield variables from freestream or flowfield conditions, based on the flow

direction at the boundary. When used at inflow boundaries, the inflow total pressure is preserved, which may result in initial Mach numbers being altered.

For some cases, this boundary condition may also be used at unconfined outflow boundaries, but the **outflow** boundary condition described in [Section 5.4.3](#) is generally recommended instead. In particular, experience has shown that using the **freestream** condition at outflow boundaries with a shear layer exiting the computational domain may result in very slow convergence, and the solution may not be very accurate. There may also be the possibility of a mixed boundary, such as a supersonic outflow with a small subsonic region in the wall boundary layer. In this case, the pressure in the supersonic region is extrapolated to the boundary from upstream, but the pressure in the subsonic region is set from the freestream value. This may cause a small disturbance at the boundary, which can be corrected by specifying the boundary as an **outflow** boundary and imposing a constant pressure for the entire boundary.

### 5.4.2 Arbitrary Inflow

The **arbitrary inflow** boundary condition allows conditions to be specified on regions of zonal boundaries where flow is entering the zone. Such a capability may be required to describe a thermally stratified nozzle input flow, or a jet emanating from a wall. The inflow profile may be specified in a number of different ways: as uniform flow, as a point-by-point (*xyz*) profile, or as uniform flow over a range of grid indices. The **ARBITRARY INFLOW** keyword is used in the input data file to indicate desired flow properties.

*Keywords:* **ARBITRARY INFLOW**

### 5.4.3 Outflow

The **outflow** boundary condition may be used for internal flows at boundaries where subsonic flow is leaving the computational domain, such as at the exit plane of an inlet, diffuser, or auxiliary flow duct. It is also recommended for downstream outflow boundaries in external flow problems, especially if a shear layer is exiting the computational domain.

Characteristic theory indicates that only one physical condition is required to define the boundary condition. One may know one of the following at the outflow boundary: mass flow, exit pressure, or exit Mach number.

The mass flow (in lb<sub>m</sub>/sec) may be specified at the outflow boundary (see the **MASS FLOW** keyword). The mass flow may be the actual or corrected value. One may alternatively specify a mass flow ratio of the mass flow through the inflow capture area specified in GMAN. During the solution, the mass flow boundary condition is applied every five iterations by default, which reduces computational costs. The integrated mass flow is compared with the desired value, and the exit static pressure is adjusted to reach the desired mass flow in the steady state. WIND displays the ratio between the computational and desired mass flows and the modified pressure at each application of the boundary condition. If you experience difficulty in converging the mass flow, you should consider setting a constant back pressure at the duct exit.

A constant exit pressure may also be specified (see the **DOWNSTREAM PRESSURE** keyword) at the outflow boundary. This option results in a very reflective boundary condition, which may cause difficulties in convergence of the solution, especially for internal flows. One alternative is to allow the exit pressure to vary spatially according to the distribution of the solution points adjacent to the boundary. This option is selected through the **VARIABLE** option of the **DOWNSTREAM PRESSURE** keyword. The **UNSTEADY** option of the **DOWNSTREAM PRESSURE** keyword may be used to specify either

a sinusoidal or user-defined pressure oscillation at an outflow boundary.

The mass-averaged Mach number may be specified at the outflow boundary using the **DOWNSTREAM MACH** keyword. This boundary condition is identical to the Chung-Cole compressor face boundary condition discussed below. It simulates the uniform Mach number characteristics that have been observed experimentally at compressor faces. This boundary condition also corresponds to a fairly uniform mass flux through the outflow boundary.

The compressor face models of Chung and Cole (1996) and of Mayer and Paynter (1994) are also available at outflow boundaries, through the **COMPRESSOR FACE** keyword. Both models are based on the observation that turbine engine conditions set the corrected mass flow, and that this corresponds directly to the average Mach number at the compressor face. These boundary conditions have been implemented mainly for the analysis of unsteady flow; however, they have also been shown to be robust for the establishment of steady-state, supercritical inlet flows.

The computational grid at the outflow boundary should be such that it is modeled with a single computational plane (constant  $i$ ,  $j$ , or  $k$ ) in a single zone. If two zones merge together near the exit, one should create a small exit zone to accommodate the outflow boundary condition.

*Keywords:* **COMPRESSOR FACE**, **DOWNSTREAM MACH**, **DOWNSTREAM PRESSURE**, **MASS FLOW**

## 5.5 Grid Topology Boundary Conditions

The **reflection**, **self-closing**, **singular axis**, and **pinwheel axis** types of boundary conditions form a group involving the simulation of the flow at topological surfaces in the grid.

### 5.5.1 Reflection

The **reflection** boundary condition simulates a plane of symmetry, and is the same as a solid, slip wall boundary condition. Therefore, within WIND, the **inviscid wall** boundary condition is actually applied. The **reflection** boundary condition type does provide a descriptive label for the boundary solution points, which may be of use to several auxiliary CFD codes which generate or use reflected grids.

### 5.5.2 Self-Closing

The **self-closing** boundary condition can be used at boundaries for which grid lines connect end-to-end (e.g.,  $i_{max}$  connecting to  $i_1$  as in an O-grid) in a point-match manner. The boundary condition simply averages the flow variables from both sides of the boundary and assigns the average to the two boundaries. This condition was formerly used for polar-type grids in duct flowfields. Note: For best results, one should consider using the **coupled** boundary condition instead of **self-closing**.

### 5.5.3 Singular Axis

The **singular axis** boundary condition is imposed at locations where an entire or part of a zonal boundary has collapsed to a line (*not a point*). Thus, along the other boundary direction, the grid points are coincident. The singular grid point is evaluated by taking the distance-weighted average of the solution of the adjacent grid points encircling the axis. This boundary condition is not to be used when the singularity line collapses to a point. For partially singular boundaries, the average is computed only over the singular portion of the boundary. Excessive use of this boundary

condition is not recommended since flow conservation is not preserved. One should attempt to use non-singular grids whenever possible. [TEST 118](#) allows the choice of which variables are averaged. [TEST 150](#) can be used to explicitly set certain velocity components to zero when the singular axis occurs on symmetry planes. [TEST 199](#) excludes the last grid point on the singular axis from being averaged.

#### 5.5.4 Pinwheel

The `pinwheel axis` boundary condition is used when there exists multiple singularities at various locations on boundaries. This boundary condition does not zero any velocity components — it simply takes the average. [TEST 118](#) allows the choice of which variables are averaged. [TEST 165](#) allows input of the integer for the order of magnitude of the tolerance for singularity (the default is  $10^{-8}$ ).

### 5.6 Zonal Interface Boundary Conditions

The `coupled` and `chimera` boundary condition types involve the interactions between zones of the domain.

#### 5.6.1 Coupled

The `coupled` boundary condition is imposed at regions where zones connect. Zone coupling is the name of the process by which WIND transfers flowfield information from one zone to another across conterminous computational grid boundaries. This process uses geometric interpolation factors stored in the grid file, which have been previously computed in the GMAN program. The zone interfaces occur at the boundaries of the zones, and so are imposed as boundary conditions at the end of each cycle of the computation.

The default zone coupling algorithm is based on Roe's flux-difference splitting scheme. The `COUPLING` keyword allows one to change a zone coupling algorithm based on one-dimensional characteristic theory.

Zone boundaries should be treated the same as you would treat interior grid planes; there should not be any large changes in grid stretching or orthogonality at the boundary. In addition, zone boundaries should not be placed in regions of strong flowfield gradients, especially horizontally along jet shear layers or aligned with normal shocks. In other words, use your best engineering judgement in placing zone boundaries in your solution.

*Keywords:* `COUPLING`

**Periodic Boundaries.** Periodic boundaries are treated as normal coupled boundaries in WIND, with the connection data stored in the common grid (`.cgd`) file when setting boundary conditions with GMAN. No keyword is needed in the input data (`.dat`) file.

GMAN's `CONNECT MODE` option is used to specify how to move one boundary surface to overlay it on the corresponding periodic boundary surface. There are two connection modes — translation and rotation. For translation, the user defines the  $\Delta x$ ,  $\Delta y$ , and  $\Delta z$  movement for the surface. For rotation the user sets the center of rotation and the angles. GMAN then moves the current boundary surface using the specified method, and attempts to couple the two surfaces together as if they were coexistent boundaries. If successful, it saves the data as normal coupling data for WIND. Thus the boundaries look connected even though they are physically separated. The periodic boundary



surfaces do not have to be in the same zone, nor do they have to be point matched. They only have to line up physically once the movement has been performed.

Specifically, to set a periodic boundary condition in GMAN:

- In graphics mode, from the main menu select **BOUNDARY COND.**
- Pick the zone and boundary for one of the two periodic boundaries.
- If the boundary condition at the boundary is not **UNDEFINED** (you may select **IDENTIFY POINTS** to check), change it to **UNDEFINED** by doing:
  - Select **MODIFY BNDY.**
  - Select **CHANGE ALL.**
  - Select **UNDEFINED.**
  - From the main menu, select **BOUNDARY COND.**
  - Select **YES-UPDATE FILE.**
- Select **MODIFY BNDY.**
- Select **COUPLE.**
- Select **SEL OTHER BND.**
- Pick the zone and boundary for the other periodic boundary. (Note that no default zone is pre-selected here, even for single-zone grids, so you must select both the zone and boundary.)
- Select **SET COUP MODE.**
- Under **CONNECT MODE**, click on the “\*\* NONE \*\*”.
- Respond to the prompts at the bottom of the screen to set the connection mode, and to specify the translation or rotation data.
- Select **CONNECT** (not **COUPLE**) to actually connect the two boundaries.

**Actuator Disk.** To simulate fan or compressor discontinuities, WIND provides an actuator disk modeling capability, which acts as a modification to the zone coupling boundary condition. The model assumes an infinitesimally thin disk and must be applied at a coupled zonal interface. The actual discontinuity is specified in the input data file as a solid-body rotation or free vortex flow, the effects of which are applied when transferring flow information between the two specified zone boundaries.

*Keywords:* [ACTUATOR](#)

**Screen.** Flowfield screens may also be modeled in WIND as discontinuities across coupled zonal interfaces. In the WIND input data file, one must specify the zones and boundaries between which the screen is located, the solidity of the screen, and one of several methods for calculating the losses through the screen. The screen model is not intended for use with choked screens, where the screen is significantly limiting the mass flow rate. During the solution start-up phase, it may be necessary to specify a low solidity, then increase it to the desired value to avoid strong choking in transients.

*Keywords:* [SCREEN](#)

### 5.6.2 Chimera

The `chimera` boundary condition indicates that the zone boundary is in an overlap region.

## 5.7 Miscellaneous Boundary Conditions

The `undefined` and `frozen` boundary condition types are included in this section, mostly because they don't fit well in any other section, and are fairly self-explanatory.

### 5.7.1 Undefined

The default boundary condition type in GMAN is `undefined`. If not set to another boundary condition type, the boundary solution point is evaluated as an average of local solution points. In practice, one should specify the actual boundary condition type for all boundary solution points and avoid having undefined boundary solution points.

If all the points on a boundary surface have an undefined boundary condition, an error message is printed and the solution will abort. If only some points have an undefined boundary condition, a warning message is printed and the solution will continue.

### 5.7.2 Frozen

The `frozen` boundary condition simply signals a boundary solution point to retain its value as read in from the initial solution file.

## 6 Convergence Monitoring

Monitoring and properly assessing convergence levels during a WIND run are critical in obtaining meaningful, useful results. WIND allows you to track convergence by following residuals and/or integrated forces, moments, and mass flow. For engineering applications, the recommended convergence monitoring method is the tracking of integrated quantities of interest. For example, if you are modeling a wing/body geometry to determine drag, you should monitor integrated drag and set some reasonable bounds on drag oscillations as your convergence criterion.

### 6.1 Residuals

In steady-state (non-time-accurate) solutions, the residuals are the amounts by which the solution vector changes in a single iteration. Ideally, in approaching a steady state solution, the residuals should approach zero. However, in practice, complex geometric and flowfield features may limit the reduction in the residuals to about two orders of magnitude. WIND prints solution residuals to the [list output file](#), in order to get a general idea of solution convergence. For each zone, WIND prints the zone number, cycle number, location of the maximum residual ( $i$ ,  $j$ , and  $k$  indices), equation number for which the maximum residual occurred, the value of the maximum residual, and the L2-norm of all the residuals for all the equations over all the points in that zone. By default, the residuals are printed each iteration. The output interval may be changed, however, using the [CYCLES](#) and [ITERATIONS](#) keywords. Residuals may be plotted by first using the auxiliary program *resplt* to generate a GENPLOT file, then using that file as input to the [plot](#) command in the [CFPOST post-processing package](#).

The L2-norm of the residuals will give you an idea of the overall convergence of the solution. The location of the maximum residual may give you insight into problems with a particular solution. After solution “bombs” or when convergence is unacceptably slow, the location of the maximum residual is the first place you should look for potential problems with the solution.

If the maximum solution residual decreases by four orders of magnitude from its maximum value in a particular zone, WIND will display a “converged” message in the list output file and stop iterating in that zone. Note that, if the maximum residual is high due to large initial solution transients, WIND may decide that a zone is converged when it may require significantly more iterations. In the input data file, you may specify the number of orders of magnitude required before WIND’s convergence check is triggered, and you may also set the absolute value of the maximum or L2-norm residual to be checked for convergence.

*Keywords:* [CONVERGE](#), [CYCLES](#), [ITERATIONS](#)

### 6.2 Integrated Flowfield Quantities

As noted above, the optimal convergence checking criterion for a solution is an integrated quantity of interest for a particular study. For example, for an afterbody drag study to calculate pressure drag, the integrated drag is obviously the best quantity to track in monitoring convergence. This may be accomplished using WIND’s flowfield integration capability. In the input data file, you may specify a number of computational surfaces for which WIND will integrate forces, moments, and/or mass flows. If you specify reference length, area, and moment center, WIND will output the forces and moments as coefficients; you may even specify output of lift and drag coefficients in place of  $x$  and  $y$  force coefficients.

You may specify zones and computational grid indices of subsets to be integrated, and you may

request that different quantities be integrated on each subset. You may also control the frequency of output and whether the code outputs subset totals and/or zonal and grand totals. The auxiliary program *resplt* will produce GENPLOT plot files of output integration quantities versus solution cycle number, which may then be used as input to the *plot* command in the **CFPOST** post-processing package.

*Keywords:* **LOADS**

### 6.3 History Tracking of Flow Data

In time-accurate calculations, where the time step is being specified in seconds and is the same throughout the flow field, you may wish to track a particular flow variable as the solution advances in time. For example, you may be interested in monitoring the static pressure on a backward-facing step as vortices roll off the back of the step. Currently, you may track only Mach number, static pressure, static temperature, and the Cartesian velocity components  $u$ ,  $v$ , and  $w$ , by requesting history tracking in the input data file. Specification must include zone numbers and ranges of grid points to track, and the frequency of the sampling. When you request history tracking, WIND creates a *time history file* containing the sampled data. The auxiliary program *timplt* must be run to extract data from the history file into GENPLOT plot files, which may then be used as input to the *plot* command in the **CFPOST** post-processing package.

*Note: The time history tracking capability does not currently work when WIND is run in parallel mode, and will cause an abort.*

*Keywords:* **HISTORY**

## 7 Files

A number of files are used by WIND in the course of a solution. The script file you run to submit WIND jobs will assign all the necessary files to their appropriate Fortran unit numbers, so you should not need to do any of that yourself. Each of the support files is described briefly below.

### 7.1 Input Data File (*.dat*)

#### Fortran unit number 5

The *input data file* is the primary control file for WIND and must be created for each case you want to run with the code. Input data and code options are entered through the use of descriptive keywords in this file. You should observe the following formatting rules in creating the input data file:

1. The first three lines of the file are reserved for geometry, flow condition, and arbitrary titles, respectively. Each of these titles may be up to 64 characters long. None of these first three lines may start with the (case-insensitive) word “Include”.
2. Comment lines, beginning with a ‘/’ character, may be placed anywhere in the file after the first three lines. Data file readability may be improved dramatically through the liberal use of comments — for example, separating logical sections of the data file: numerical algorithm, force integrations, test options, etc.
3. Block data, such as that specified in arbitrary inflow or chemistry cases, must be contiguous — only keywords corresponding to the block may reside between the beginning and ending block indicators (e.g., **CHEMISTRY** and **ENDCHEMISTRY**).
4. Keywords may be entered in upper or lower case.
5. Abbreviations for keywords may be used, as long as they are unique. If they are not, you may not get the results you expect. For example, it’s not a good idea to use single-letter abbreviations for keywords.

The following is an example of a simple input data file:

```
Geometric Title
Flowfield Condition Title
Optional Title
Freestream static 0.9 14.7 530. 4. 0.
Cycles 15
Inviscid
End
```

### 7.2 Grid File (*.cgd*)

#### Fortran unit number 11

The computational grid used by WIND for a particular case is stored in the *grid file*. In this file are stored the  $(x, y, z)$  coordinates of all computational grid points, zone coupling interpolation factors, and grid reference and scaling data.

This file was originally referred to as the *common* grid file, so named because the file was formatted according to Boeing's Common File Format (CFF). Starting with version 4.0 however, WIND also supports grid files in CGNS (CFD General Notation System) format.<sup>16</sup> Both common files and CGNS files are binary, and portable to virtually every hardware platform, except Cray, with no need for explicit data conversions.

Grid files for WIND may be created by several mesh-generation codes in either common file or CGNS format. If necessary, the *cfenvt* utility may be used to convert a variety of other formats, including PLOT3D xyz format, to common file format.

Zone coupling, reference, and scaling data are added to common grid files using the **GMAN program**. Common grid files are also used in the **CFPOST post-processing package**. Neither GMAN nor CFPOST currently support CGNS files, however.

*Keywords:* CGNSBASE

### 7.3 Flow File (.cfl)

Fortran unit number 20

The *flow file* contains the computed flow field. For Navier-Stokes and Euler solutions, the file contains density, momentum, and energy data, and, for viscous solutions, turbulence data. The flow file also records the current solution cycle number to allow the file to be used for solution restarts.

Like the grid file, the flow file was originally referred to as the *common* flow file, but may now be written in either common file or CGNS format. Several graphical post-processing programs are able to read both common files and CGNS files.

The **CFPOST post-processing package** may be used with common flow files to produce other files for post-processing and/or to create flowfield plots directly. CFPOST does not currently support CGNS files, however.

*Keywords:* CGNSBASE

### 7.4 Global Newton File (.cfk)

Fortran unit number 16

The *Global Newton file* is a copy of the flow file used for Global Newton time marching. It holds the current sub-iteration of the time step. It is only created when Global Newton time marching is active, and is a scratch file. If the code aborts, the solution restarts from the conditions in the flow file, at the beginning of the time level, so the Global Newton file does not need to be saved.

### 7.5 Boundary Data File (.tda)

Fortran unit number 14

WIND's *boundary data file* is used during solution restarts, and results in smoother restarts, especially with higher-order boundary coupling. The file acts as a buffer for the transfer of zone coupling information and a holding bin for data needed throughout a WIND run but not stored in the grid or solution file.

---

<sup>16</sup>Detailed information on the CGNS standard may be found at the CGNS web site, at <http://www.cgns.org/>.

## 7.6 Time History File (*.cth*)

Fortran unit number 19

The *time history file* is a common file which stores data resulting from the use of WIND's history tracking capability. The file contains a lookup table corresponding to the range of computational indices tracked during the run, time stamp data, and flowfield data for each time stamp. The file name will actually have the suffix *.cth.DATE*, where *DATE* is the date and time the job was started. Upon completion of a time history run, the auxiliary program *timplt* may be used to view the contents of the time history file.

*Note:* An improved time history tracking capability was added to WIND alpha 5.52. This change also required some changes to the shell scripts used to run the WIND code. With the new scripts, the *DATE* suffix to the file name has been eliminated.

*Keywords:* HISTORY

## 7.7 List Output File (*.lis*)

Fortran unit number 6

The purpose of WIND's *list output file* is to echo the input from the input data file, track convergence and integration results, record CPU/job statistics, and log code error messages. The auxiliary program *resplt* may be used to extract convergence and integration data from the list output file and create a GENPLOT-style data file, which may be plotted with the *plot* command in the *CFPOST post-processing package*. For parallel-processing runs, this file also contains messages relating to the PVM system and to the allocation and operation of slave processors. Convergence data is written to this file for each of the equation sets solved during the run, including the Euler, Navier-Stokes, and non-algebraic turbulence model sets.

## 7.8 WIND Stop File (*NDSTOP*)

Fortran unit number 1

The *stop file* is used to stop WIND execution cleanly in the middle of a CFD solution. Although a solution may be stopped simply by killing the WIND process from the system queue, doing so will not ensure a clean update of all zonal flowfield data to the flowfield file. The stop file — named *NDSTOP* — provides a means to cleanly shut down a running solution, completing the current cycle or zone, performing zone coupling, updating the flowfield file, and removing all symbolic links to Fortran file unit numbers.

To stop the code, you must place one of two words in the stop file in the directory in which the WIND job is running. The word *STOP* in the stop file signals WIND to complete the current cycle (at the end of the last zone) and exit. In single-processing mode, you may also use the word *STOPZONE* in the stop file to stop WIND after completing the zone currently in memory.<sup>17</sup> You may then restart the code in your next run, starting at the next zone or back at zone one. Regardless of the existence of this file, WIND will stop if the requested number of cycles has been computed, or if your solution has converged.

During the clean-up procedure at the end of the job, the *NDSTOP* file is automatically removed.

---

<sup>17</sup> In multi-processing mode, *STOPZONE* does the same thing as *STOP*.

Note that the directory in which the WIND job is running, where the *NDSTOP* file must be, is not necessarily the one you were in when the *wind* command was issued. If you don't use the *-runinplace* option to the *wind* script (see [Section 8.1](#)), the job will be run in a remote directory. The root name of the remote run directory may be specified using the *-runroot* option, or in response to a command line prompt. The default for the "root name" of the remote directory is */tmp*. The full name of the remote directory will be *rootname/userid/basename.scr*, where *rootname* is the root name described above, *userid* is your userid, and *basename* is the base name of your *.dat* file.

On a Unix system, you might submit a simple 'at' job for a later time as follows:

```
at 0530 monday
echo STOP > NDSTOP
^D
```

At 5:30 AM on the next Monday, the system would create the *NDSTOP* file with the word 'STOP' in it. Note that this will not stop the run exactly at 5:30 AM; WIND must still complete the current cycle, which may take an hour or more for large cases.

Keywords: [RESTART](#)

## 7.9 Temperature and Transition Specification Files

### Fortran unit number 45

WIND can read temperature and transition files, specifying the temperature or transition to turbulence on any boundary surface, that were created with the older *tmptrn* utility. This option is included for backward compatibility with WIND 2.0, and is only needed for an initial (i.e., non-restart) run. The *wind* script copies the files to the run directory, and WIND opens the files directly (unit 45) using the file names specified with the [TTSPEC](#) keyword.

New applications should use the latest *tmptrn* utility to write the temperature/transition data into the common flow (*.cfl*) file directly.

Keywords: [TTSPEC](#)

## 7.10 Chemistry Files (*.chm*)

### Fortran unit number 26

The generalized chemistry files contain all the information WIND requires to compute a general chemistry mixture. A chemistry file has a [header line](#), followed by three sections containing [thermodynamic coefficients](#), [finite rate coefficients](#), and [transport properties](#).

### 7.10.1 Header

The header line must contain a single parameter, *ispec*, specifying the type of reaction rate and the format for the finite rate data. The format for the header line is<sup>18</sup>:

|              |       |
|--------------|-------|
| <i>ispec</i> | ISPEC |
|--------------|-------|

<sup>18</sup>The parameter *ispec* is read using a list-directed read statement. The ISPEC label following the value is thus optional, but present in the standard chemistry files supplied with the WIND code. The labels following the parameters *ns*, *nreq*, *ndeq*, and *tfrmin*, described in [Section 7.10.2](#) and [Section 7.10.3](#), are also optional.



where

| <i>ispec</i>   | <u>Reaction Rate and Format</u>                                           |
|----------------|---------------------------------------------------------------------------|
| 1, 4, 100, 110 | Forward and backward rates from equilibrium constant polynomial, Format 1 |
| 3, 130         | Forward and backward rates, Format 2                                      |
| 120            | Forward rate only, Format 3                                               |

### 7.10.2 Thermodynamic Coefficients

This section of the chemistry data file contains the information necessary to compute the specific heat  $c_p$  for each species. The value of  $c_p$  is defined by a series of polynomials, each valid within a defined temperature range, of the form:

$$c_p = a_1 + a_2T + a_3T^2 + a_4T^3 + a_5T^4$$

where  $T$  is in K, and  $c_p$  is in J/kg-K.

The format for this section is:

```

THERMODYNAMIC COEFFICIENTS
Title, line 1
Title, line 2
ns      NS
Information defining species 1
...
Information defining species ns
```

where  $ns$  is the number of species.

For each species, the information defining the species and for computing  $c_p$ , etc., is read as follows:

```

      read (26,100) name,ncurves,ds1,hm1,ds2,hm2,unk1,unk2,con,molwt
      do 10 ic = 1,ncurves
        read (26,110) t1,t2,(a(i,ic),i=1,5),shf(ic),F(ic)
    10 continue
    100 format (a8,7x,i5,4x,2(a2,f3.0),16x,f10.1,5x,f10.3)
    110 format (5e15.5)
```

where

| <u>Variable</u>  | <u>Definition</u>                                                                                                                   |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| <b>name</b>      | Name of species                                                                                                                     |
| <b>ncurves</b>   | Number of temperature segments in $c_p$ vs. $T$ curve                                                                               |
| <b>ds1</b>       | Name of first constituent atomic species in <b>name</b>                                                                             |
| <b>hm1</b>       | Number of atoms per molecule of <b>ds1</b>                                                                                          |
| <b>ds2</b>       | Name of second constituent atomic species                                                                                           |
| <b>hm2</b>       | Number of atoms per molecule of <b>ds2</b>                                                                                          |
| <b>unk1</b>      | Unknown (0.0)                                                                                                                       |
| <b>unk2</b>      | Unknown (0.0)                                                                                                                       |
| <b>con</b>       | Coefficient defining number of degrees of freedom, defined as (translational degrees of freedom + rotational degrees of freedom)/2. |
| <b>molwt</b>     | Molecular weight                                                                                                                    |
| <b>t1,t2</b>     | Beginning and end of temperature range for curve segment <b>ic</b>                                                                  |
| <b>a(1-5,ic)</b> | Polynomial coefficients $a_1$ – $a_5$ for curve segment <b>ic</b>                                                                   |
| <b>shf(ic)</b>   | Heat of formation of species, in J/kg                                                                                               |
| <b>F(ic)</b>     | Gibb's free energy (for future use in gas constant)                                                                                 |

### 7.10.3 Finite Rate Coefficients

This section of the file contains the reaction rate information. There are three possible formats for this section, depending on the value of *ispec*.

**Format 1; *ispec* = 1, 4, 100–119.** This option may be used for any chemically reacting flow. Information is specified defining the forward reaction rate  $k_f$  and the equilibrium constant  $K$ . The rates themselves are computed using the equations

$$k_f = CT^S e^{-D/(\kappa T)} \quad (1)$$

$$k_b = k_f/K \quad (2)$$

$$K = \exp(a_1 + a_2 Z + a_3 Z^2 + a_4 Z^3 + a_5 Z^4) \quad (3)$$

$$Z = 10,000/T \quad (4)$$

where  $T$  is the temperature in K, and  $\kappa$  is the Boltzman constant.

The various parameters and polynomial coefficients are read from the chemistry data file, in the following form:

```

FINITE RATE COEFFICIENTS
Title, line 1
Title, line 2
nreq ndeq      NREQ, NDEQ
tfrmin         TFRMIN
Information defining dissociation reaction 1
...
Information defining dissociation reaction ndeq
Information defining exchange reaction 1
...
Information defining exchange reaction nreq - ndeq

```

where *nreq* is the total number of reactions (dissociation + exchange + ionization), *ndeq* is the number of dissociation reactions (i.e., have a third body), and *tfrmin* is the temperature in K below which no reactions occur.

For each reaction, the information defining the reaction and for computing the reaction rates is read as follows:

```

      read (26,100) r1,r2,p1,p2,S,D,(a(i),i=1,5),nthd
      do 10 i = 1,nthd
        read (26,110) spec,C
      10 continue
    100 format (4(a5,3x),2e12.4/5e12.4,i4)
    110 format (16x,a8,e12.4)

```

where

| <u>Variable</u> | <u>Definition</u>                                                                                                                                  |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>r1</b>       | Name of reactant 1                                                                                                                                 |
| <b>r2</b>       | Name of reactant 2. (Leave blank for dissociation reactions.)                                                                                      |
| <b>p1</b>       | Name of product 1                                                                                                                                  |
| <b>p2</b>       | Name of product 2                                                                                                                                  |
| <b>S</b>        | Temperature exponent <i>S</i> in equation (1)                                                                                                      |
| <b>D</b>        | <i>D</i> / $\kappa$ in equation (1)                                                                                                                |
| <b>a(1-5)</b>   | Polynomial coefficients $a_1$ – $a_5$ in equation (3)                                                                                              |
| <b>nthd</b>     | Number of third bodies, implies <b>nthd</b> reactions (one for each third body). Always set <b>nthd</b> = 0 for exchange and ionization reactions. |
| <b>spec</b>     | Name of third body reactant (not needed if <b>nthd</b> = 0 or 1)                                                                                   |
| <b>C</b>        | Third body coefficient for species <b>spec</b> as third body, or average if <b>nthd</b> = 0 or 1                                                   |

With this option, the data is read in subroutine **frtin1**, and the rates are computed in **rates1** (for *ispec* = 1), **rates4** (*ispec* = 4), **rates** (*ispec* = 100), or **ratesa** (*ispec* = 110). [At some point, the special-case options *ispec* = 1 and 4 may be eliminated in favor of one of the more general *ispec* = 100–119 options.]

**Format 2; *ispec* = 3, 130–139.** This option is similar to the previous one, except that the forward and backward reaction rates are computed separately, using the equations

$$k_f = C_f T^{S_f} e^{-D_f/(\kappa T)} \quad (5)$$

$$k_b = C_b T^{S_b} e^{-D_b/(\kappa T)} \quad (6)$$

where *T* is the temperature in K, and  $\kappa$  is the Boltzman constant.

The various parameters are read from the chemistry data file, in the following form:

```

FINITE RATE COEFFICIENTS
Title, line 1
Title, line 2
nreq ndeq      NREQ, NDEQ
tfrmin         TFRMIN
Information defining dissociation reaction 1
...

```

*Information defining dissociation reaction ndeq*  
*Information defining exchange reaction 1*  
 ...  
*Information defining exchange reaction nreq - ndeq*

where *nreq* is the total number of reactions (dissociation + exchange + ionization), *ndeq* is the number of dissociation reactions (i.e., have a third body), and *tfrmin* is the temperature in K below which no reactions occur.

For each reaction, the information defining the reaction and for computing the reaction rates is read as follows:

```

      read (26,100) r1,nr1,r2,nr2,p1,np1,p2,np2,Sf,Df,Cf
      read (26,110) Sb,Db,Cb
100  format (4(a5,f3.1),3e12.4)
110  format (32x,3e12.4)

```

where

| Variable | Definition                                                               |
|----------|--------------------------------------------------------------------------|
| r1,nr1   | Name and number of reactant 1                                            |
| r1,nr2   | Name and number of reactant 2. (Leave blank for dissociation reactions.) |
| p1,np1   | Name and number of product 1                                             |
| p2,np2   | Name and number of product 2                                             |
| Sf,Sb    | Temperature exponents <i>Sf</i> and <i>Sb</i> in equations (5) and (6)   |
| Df,Db    | $D_f/\kappa$ and $D_b/\kappa$ in equations (5) and (6)                   |
| Cf,Cb    | Third body coefficients $C_f$ and $C_b$ in equations (5) and (6)         |

With this option, the data is read in subroutine **frtin3**, and the rates are computed in **rates3** (for *ispec* = 3), or **ratesb** (*ispec* = 130). [At some point, the special-case option *ispec* = 3 may be eliminated in favor of one of the more general *ispec* = 130–139 options.]

**Format 3; *ispec* = 120–129.** This option is only for global 1- or 2-reaction chemistry, and only for forward reactions. It is a quick method for simulating detonation problems, for example, in which reactions proceed only forward to completion. The forward reaction rates are computed using the equation

$$k = CT^S e^{-D/(\kappa T)} \quad (7)$$

where *T* is the temperature in K, and  $\kappa$  is the Boltzman constant. There are no backward reactions.

The various parameters and polynomial coefficients are read from the chemistry data file, in the following form:

```

FINITE RATE COEFFICIENTS
Title, line 1
Title, line 2
nreq ndeq      NREQ, NDEQ
tfrmin         TFRMIN
Information defining reaction 1
...
Information defining reaction nreq

```

where *nreq* is the total number of reactions (dissociation + exchange + ionization), *ndeq* is the number of dissociation reactions (i.e., have a third body), and *tfrmin* is the temperature in K below which no reactions occur.

For each reaction, the information defining the reaction and for computing the reaction rates is read as follows:

```

      read (26,100) r1,nr1,r2,nr2,p1,np1,p2,np2,S,D,(a(i),i=1,5),nm
      read (26,110) Cb1,Cb2,C
100 format (4(a5,f3.1),2e12.4/5e12.4,i4)
110 format (2f8.3,8x,e12.4)

```

where

| <u>Variable</u> | <u>Definition</u>                                     |
|-----------------|-------------------------------------------------------|
| <b>r1,nr1</b>   | Name and number of reactant 1                         |
| <b>r2,nr2</b>   | Name and number of reactant 2                         |
| <b>p1,np1</b>   | Name and number of product 1                          |
| <b>p2,np2</b>   | Name and number of product 2                          |
| <b>S</b>        | Temperature exponent <i>S</i> in equation (7)         |
| <b>D</b>        | $D/\kappa$ in equation (7)                            |
| <b>a(1-5)</b>   | Place holder for use with future <i>ispec</i> options |
| <b>nm</b>       | Place holder for use with future <i>ispec</i> options |
| <b>Cb1,Cb2</b>  | Exponents on concentrations of reactants 1 and 2      |
| <b>C</b>        | Third body coefficient <i>C</i> in equation (7)       |

With this option, the data is read in subroutine **frtin2**, and the rates are computed in **ratesf**.

#### 7.10.4 Transport Coefficients

This section of the chemistry data file contains the coefficients used to compute the laminar viscosity and thermal conductivity. The formulation is based on Wilke's law which uses Sutherland's law individually for each species. Thus for each species, there are equations of the form:

$$\frac{\mu}{\mu_0} = \left( \frac{T}{T_{0\mu}} \right)^{\frac{3}{2}} \frac{T_{0\mu} + S_{\mu}}{T + S_{\mu}} \quad (8)$$

$$\frac{\kappa}{\kappa_0} = \left( \frac{T}{T_{0\kappa}} \right)^{\frac{3}{2}} \frac{T_{0\kappa} + S_{\kappa}}{T + S_{\kappa}} \quad (9)$$

The various parameters and polynomial coefficients are read from the chemistry data file, in the following form:

```

TRANSPORT COEFFICIENTS
Title, line 1
Title, line 2
Information for species 1
...
Information for species ns

```

For each species, the information is read as follows:

```

      read (26,100) name,ncurves,t1,t2,mu0,t0mu,smu

```

```

      read (26,110) k0,t0k,sk
    do 10 ic = 2,ncurves
      read (26,120) t1,t2,mu0,t0mu,smu
      read (26,110) k0,t0k,sk
    10 continue
100 format (a8,2x,i5,2f10.3,1x,3e12.4)
110 format (36x,3e12.4)
120 format (15x,2f10.3,1x,3e12.4)

```

where

| <u>Variable</u> | <u>Definition</u>                                                  |
|-----------------|--------------------------------------------------------------------|
| <b>name</b>     | Name of species                                                    |
| <b>ncurves</b>  | Number of temperature segments in $\mu$ and $\kappa$ vs. $T$ curve |
| <b>t1,t2</b>    | Beginning and end of temperature range for curve segment <b>ic</b> |
| <b>mu0</b>      | Reference viscosity                                                |
| <b>t0mu</b>     | Reference temperature in equation (8)                              |
| <b>smu</b>      | Reference temperature offset in equation (8)                       |
| <b>k0</b>       | Reference conductivity                                             |
| <b>t0k</b>      | Reference temperature in equation (9)                              |
| <b>sk</b>       | Reference temperature offset in equation (9)                       |

*Keywords:* [CHEMISTRY](#)

## 7.11 Reserved Files

Unit numbers 15, 21, and 55 are reserved for proprietary features, and should not be used by WIND developers.

## 8 Scripts

WIND is run using a Unix script which attaches appropriate files and either starts WIND interactively or submits the job to a queueing system. In addition, this script checks for special files and may be used to re-invoke WIND after performing intermediate operations.

### 8.1 *wind* — Run WIND Code

The usual procedure for running the WIND code is via a command-line script called *wind*. With this approach, the specific executable to be run (e.g., production version, alpha version, etc.), the names of the required input and output files, etc., may be defaulted, specified on the command line, or entered as responses to prompts. The various options to the script are described below in the form of a Unix-style man page for *wind*.

If the *Tcl/Tk* interpreter is available, the WIND code may be launched via an interactive GUI by using **-g** as the first option to the *wind* script, i.e., by typing **wind -g**. In this case, the executable to be run, the file names, etc., are specified via the GUI. Any options to the *wind* script following the **-g** are ignored. If the **-g** option is specified, but the *Tcl/Tk* interpreter is not available, the usual command-line interface will be used.

#### The *wind* Man Page

##### NAME

*wind* — a computational platform for solving various sets of equations governing fluid dynamics

##### SYNOPSIS

```
wind [-batch] [-betime time] [-binroot directory] [-cfdrootrem directory] [-dat datfile]
[-debug] [-debugger debugger] [-endtime time] [-flow cflfile] [-g] [-grid cgdfile]
[-grpcharge number] [-help] [-list lisfile] [-memory number] [-mp] [-mpmode mode]
[-ncpu number] [-nice_val number] [-nobg] [-(no)parallel] [-not_nice] [-nzones number]
[-program windname] [-queue_name name] [-remoterun] [-runinplace] [-runque queue]
[-runroot directory] [-tmpdir directory] [-usessh] [-walltime time]
```

##### DESCRIPTION

*wind* is a Bourne shell script used to run the WIND code, a computational platform for numerically solving various sets of equations governing physical phenomena. Currently, the code supports the solution of the Euler and Navier-Stokes equations of fluid mechanics, along with supporting equation sets governing turbulent and chemically reacting flows. See the *WIND User's Guide* for detailed information on the operation and use of the WIND code.

Internally, the *wind* script invokes other scripts, and creates on the fly a *job file* (with the suffix *.job*), which is the actual script submitted to the queue specified by the user.

##### OPTIONS

###### **-batch**

Run in “batch” mode, with all necessary input specified via command-line options. The user must specify the WIND executable to be run using the **-program** option; all the input/output file names using the **-dat**, **-list**, **-grid**, and **-flow** options; the system run queue using the **-runque** option. The user must also specify either the **-runinplace** option, or the directory to run in using the **-runroot** option. In addition, if a multi-processing control (*.mpc*) file exists, either the **-parallel** or **-noprogram** option must be specified. And, if an *.mpc* file

|                                     |                                                                                                                                                                                                                                                                                                                                                                                                          |
|-------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                     | exists and MPI message passing is being used, the number of zones must be specified using the <b>-nzones</b> option. If any of these items are missing, an error message will be displayed and execution will stop.                                                                                                                                                                                      |
| <b>-begtime</b> <i>time</i>         | The beginning time for a standard Unix <i>at/batch</i> job (i.e., when using <b>-runque</b> AT_QUE).                                                                                                                                                                                                                                                                                                     |
| <b>-binroot</b> <i>directory</i>    | Root location of the WIND executable on the system on which it will be running.                                                                                                                                                                                                                                                                                                                          |
| <b>-cfdrootrem</b> <i>directory</i> | The “CFDROOT” directory on the system on which the WIND executable will be running. The default is the same as the “CFDROOT” directory on the local machine.                                                                                                                                                                                                                                             |
| <b>-dat</b> <i>datfile</i>          | The WIND input data ( <i>.dat</i> ) file, entered without the extension (e.g., <i>wing</i> , not <i>wing.dat</i> ).                                                                                                                                                                                                                                                                                      |
| <b>-debug</b>                       | Run WIND in debug mode, using the debug package specified by the <b>-debugger</b> option. The job will automatically be run interactively, and the <b>-runinplace</b> option will be set.                                                                                                                                                                                                                |
| <b>-debugger</b> <i>debugger</i>    | Set the debug package to be used when running in debug mode. The default depends on the computer system being used.                                                                                                                                                                                                                                                                                      |
| <b>-endtime</b> <i>time</i>         | Ending time for a Unix <i>at/batch</i> job, or time to run in seconds for a job running under the basic QSUB, PBS, or LSF queueing system. For the PBS and LSF queueing systems this is the total time for the entire job, minus some length of time for termination processing (i.e., updating the flow ( <i>.cfl</i> ) file, terminating worker processes, etc.) Also see the <b>-walltime</b> option. |
| <b>-flow</b> <i>cflfile</i>         | The WIND flow ( <i>.cfl</i> ) output file, entered without the extension (e.g., <i>wing</i> , not <i>wing.cfl</i> ). If the file exists, the solution will restart using the conditions in the file as initial conditions.                                                                                                                                                                               |
| <b>-g</b>                           | When specified as the first option, use the <i>Tcl/Tk</i> interactive GUI to specify input parameters, ignoring any other options. If <b>-g</b> is not the first option specified, it is ignored.                                                                                                                                                                                                        |
| <b>-grid</b> <i>cgdfile</i>         | The WIND grid ( <i>.cgd</i> ) input file, entered without the extension (e.g., <i>wing</i> , not <i>wing.cgd</i> ).                                                                                                                                                                                                                                                                                      |
| <b>-grpcharge</b> <i>number</i>     | Group charge number for the job, used for NAS systems.                                                                                                                                                                                                                                                                                                                                                   |
| <b>-help</b>                        | Display the list of command-line options, and quit, ignoring any other options.                                                                                                                                                                                                                                                                                                                          |
| <b>-list</b> <i>lisfile</i>         | The WIND list output ( <i>.lis</i> ) file, entered without the extension (e.g., <i>wing</i> , not <i>wing.lis</i> ). Set the parameter <i>lisfile</i> to “screen” to display the list output file at the terminal.                                                                                                                                                                                       |
| <b>-memory</b> <i>number</i>        | Amount of run-time memory to request for a job running under the PBS (in megabytes) or LSF (in kilobytes) batch system.                                                                                                                                                                                                                                                                                  |
| <b>-mp</b>                          | Run using a multi-processor machine (i.e., a single machine with multiple processors, like the CRAY or SGI Origin-2000), using either PVM or MPI message passing.                                                                                                                                                                                                                                        |



|                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|---------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-mpmode</b> <i>mode</i>      | Message passing mode: either PVM or MPI. MPI may only be used on multi-processor systems. Note that the pre-compiled executables available through IVMS were built without MPI message passing. To use <b>-mpmode MPI</b> , MPI must be pre-installed on your system (unlike PVM, MPI is not distributed with WIND), and you'll need an executable that includes links to the MPI library. See the section “ <b>Installing the Build Distribution</b> ” in the <i>WIND Installation Guide</i> for instructions on creating the executable. The default message passing mode is PVM. |
| <b>-ncpu</b> <i>number</i>      | Number of CPUs to request for a job running under the PBS or LSF batch system.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>-nice_val</b> <i>number</i>  | The “nice” value (a number from 1 to 20) to use with the Unix <i>nice</i> command. Larger values cause the job to run at a lower CPU scheduling priority to lessen the impact on other jobs on the system. For interactive jobs ( <b>-runque REAL</b> ), the value is automatically set to 2. For jobs run in parallel mode, the master task, and worker tasks being run on the same system as the master, automatically run at top priority (i.e., without using <i>nice</i> ).                                                                                                    |
| <b>-nobg</b>                    | Jobs in the interactive queue normally run in the foreground if the output is being displayed on the screen, and in the background if the output is being sent to the <i>.lis</i> file. This option specifies that the interactive job is to be run in the foreground, even if the output is being sent to the <i>.lis</i> file.                                                                                                                                                                                                                                                    |
| <b>-(no)parallel</b>            | Run in parallel mode (or serial mode if <b>-noparallel</b> is specified). For interactive and <i>at/batch</i> jobs, parallel mode requires a multi-processing control ( <i>.mpc</i> ) file. If this option is not specified, and an <i>.mpc</i> file exists, the user will be asked if parallel mode should be used.                                                                                                                                                                                                                                                                |
| <b>-not_nice</b>                | Run without using the Unix <i>nice</i> command.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>-nzones</b> <i>number</i>    | Number of zones, used in MPI message passing mode on multi-processor systems.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>-program</b> <i>windname</i> | The name of the WIND executable to be used.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>-queue_name</b> <i>name</i>  | The name of the specific queue in which the job is to run. The form of the <i>name</i> parameter depends on the queueing system being used, as specified using the <b>-runque</b> option.                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>-remoterun</b>               | Bypass some of the built-in error checking for valid path names preventing error messages when running on a remote system. The “CFD-ROOT” directory on the remote machine should be specified using the <b>-cfdrootrem</b> option, and the directory to run in should be specified using the <b>-runroot</b> option.                                                                                                                                                                                                                                                                |
| <b>-runinplace</b>              | Run in place, i.e., in the directory in which the <i>wind</i> script was invoked. The default is to run in a remote directory.                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>-runque</b> <i>queue</i>     | Queueing system in which the WIND executable is to run. The parameter <i>queue</i> must be “REAL” for a real-time interactive job; “AT_QUE” to run using a standard Unix <i>at</i> or <i>batch</i> queue; “QSUB_QUE” to run                                                                                                                                                                                                                                                                                                                                                         |

in a basic QSUB queue; “QSUB\_PBS\_QUE” to run in a QSUB queue using the PBS batch system; or “BSUB\_QUE” to run in a BSUB queue using the LSF batch system.

|                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-runroot</b> <i>directory</i> | Root directory under which the WIND executable is to be run. The full name will be <i>directory/userid/basename.scr</i> , where <i>userid</i> is your userid and <i>basename</i> is the base name of your <i>.dat</i> file. The default <i>directory</i> , if neither <b>-runroot</b> or <b>-runinplace</b> is specified, is <i>/tmp</i> .                                                                                                                                                                                  |
| <b>-tmpdir</b> <i>directory</i>  | The directory for storing temporary files during a WIND run. The default is the directory specified using the <b>-runroot</b> option when running in a remote system, and the current directory when using the <b>-runinplace</b> option.                                                                                                                                                                                                                                                                                   |
| <b>-usessh</b>                   | Use <i>ssh</i> remote shell/copy commands. Note that the pre-compiled executables available through IVMS were built assuming <i>rsh</i> ( <i>remsh</i> for HP and Cray systems) will be used as the “remote shell” command for parallel operation. If your system requires the use of <i>ssh</i> , you’ll need an executable that includes links to an “ <i>ssh</i> version” of PVM. See the section “Installing the Build Distribution” in the <i>WIND Installation Guide</i> for instructions on creating the executable. |
| <b>-walltime</b> <i>time</i>     | Total wall clock time in seconds for a job running under the PBS or LSF queueing system. This includes the length of time for termination processing (i.e., updating the flow ( <i>.cfl</i> ) file, terminating worker processes, etc.) Also see the <b>-endtime</b> option.                                                                                                                                                                                                                                                |

## NOTES

Please send documentation suggestions/corrections to [towne@grc.nasa.gov](mailto:towne@grc.nasa.gov). Questions concerning the WIND code itself or the NPARC Alliance should be sent to [nparc-support@info.arnold.af.mil](mailto:nparc-support@info.arnold.af.mil)

## 8.2 *wind\_post* — Perform Post-Processing

At the end of a non-debug job that completes successfully, if a script named *wind\_post* exists in the current directory, and if the job is not being run interactively, the *wind\_post* script is invoked, as follows.

```
wind_post datname lisname cgdname cflname iter run_command
```

The six arguments are defined as:

| <u>Argument</u>    | <u>Definition</u>                                                                                                                                                                                                                                                                                                                                       |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>datname</i>     | The base name (i.e., without the three-letter extension) of the input data ( <i>.dat</i> ) and job files                                                                                                                                                                                                                                                |
| <i>lisname</i>     | The base name of the list output ( <i>.lis</i> ) file                                                                                                                                                                                                                                                                                                   |
| <i>cgdname</i>     | The base name of the grid ( <i>.cgd</i> ) file. (If CGNS files are being used, note that this is the base name of the file itself, not the name of the <code>CGNSBase_t</code> node in the file.)                                                                                                                                                       |
| <i>cflname</i>     | The base name of the flow ( <i>.cfl</i> ) file                                                                                                                                                                                                                                                                                                          |
| <i>iter</i>        | The “job iteration” number. This is a count of the number of jobs, not time step iterations. It is initialized to zero, and incremented by one before each invocation of <i>wind_post</i> . Note that the value of <i>iter</i> is not saved between runs. The next invocation of the top-level <i>wind</i> script will re-initialize the value to zero. |
| <i>run_command</i> | The command used to submit the job                                                                                                                                                                                                                                                                                                                      |

The *wind\_post* script itself must be supplied by the user. Users with some experience in writing shell scripts may use this feature to automatically post-process results at the end of a job, or to save and/or process interim results from a lengthy calculation and submit a new job.<sup>19</sup>

An example *wind\_post* Bourne shell script called *wind\_post.sh* is supplied with the WIND application distribution in the directory *wind/bin*. (There’s also an example C shell version called *wind\_post.csh*.) The example script resubmits the job with a different input file for each job iteration. It looks for an input data file named *datname.dat.nextiter*, where *datname* is the base name of the *.dat* file and *nextiter* is the *next* job iteration number (i.e., one more than the value of the input argument *iter*). The new input data file, if it exists, is *copied over* the original input data file and the job file is resubmitted.

A typical method for using this capability would be to create files named, for example, *test.dat.1*, *test.dat.2*, *test.dat.3*, etc. Then, copy *test.dat.1* to *test.dat* and submit the job as usual. By keeping the initial input file in *test.dat.1* and copying it to *test.dat*, the original initial input file is preserved.

After the initial job completes, *wind\_post* will be called with *iter* = 1. The *next* job iteration number is thus 2. The file *test.dat.2* will be copied into *test.dat*, and the job will be resubmitted, restarting WIND where it left off. If the file *test.dat.nextiter* doesn’t exist, all the input data files have been used. The job file is then removed, and the run ends.

Note the distinction between a *run* and a *job*, and the value of the job iteration number. By *run*, we mean here a single invocation by the user of the *wind* script. Within that run, multiple *jobs* may be submitted using the *wind\_post* feature, each with a different, automatically-incremented job iteration number. As noted above, the job iteration number is not saved between runs. The next time the *wind* script is invoked, starting a new run, the job iteration number will be re-initialized to zero. If it’s necessary to save the value between runs, the user-supplied *wind\_post* script should save it to a file, to be read during the next run.

The example *wind\_post* script does not do any post-processing, but that can easily be added. For example, **CFPOST** could be run, with input redirected from a command file, to create plot files or reports. Or, an interim flow (*.cfl*) file could be saved by copying it into a file with the *iter* variable as part of the file name.

---

<sup>19</sup>Note that a similar capability may be invoked using the **SPAWN** keyword in WIND’s input data file.

### 8.3 *windmp* — Run on Multi-Processor

The *windmp* script may be used as a shortcut when running WIND on a multi-processor machine (i.e., a single machine with multiple processors, like the CRAY or SGI Origin-2000). It simply invokes the *wind* script with the options *-mp*, *-runinplace*, and *-not\_nice*.

### 8.4 *windver* — Get WIND Version Number

The alpha version of WIND, and to a lesser extent the beta version, change quite frequently. Even the released production version is sometimes modified, as a result of bug fixes. Changes are summarized in the “Source History” file, accessed through the [Internet Version Management System \(IVMS\)](#).

The *windver* script may be used to get the complete version number of a WIND executable, and (for WIND 3.0 and higher) its supporting libraries. For example,

```
% windver
Versions in ./SGI6.5/R10000/bin

                Select the desired version

0: Exit wind
1: alpha version
2: Version 2.0
3: Version 3.0
4: Version 4.0

Enter number or name of executable.....[4]: 4
WIND - Version 4.115 (Last changed 2001/04/14 16:29:26)
LIBCFD - Version 2.0.20.1 (Last changed 2000/08/23 16:03:58)
LIBADF - Version 2.0.12.1 (Last changed 2001/04/04 15:21:04)
PVM - Version 3.4.3
```

*windver* lists the various WIND executables available, and prompts the user to select one of them. It then prints the complete version number, plus the date it was created, for the WIND code, the CFD\_wind library (common file processing routines, system-dependent routines, and utility routines), the ADF library (ADF routines from the CGNS project), and the PVM library (PVM routines used for parallel processing). In the above example, WIND Version 4.0 is actually Version 4.115, created on Apr 14, 2001.

By default, *windver* looks for the WIND executables in the directory specified by the environment variable `WIND_BINROOT`. If there is no environment variable of that name, *windver* will look in `CFDROOT`. The user may also specify the root location of the WIND executable using the option *-binroot*. I.e., invoking *windver* as

```
windver -binroot /usr/local/wind/wind
```

will tell *windver* that the executables are located in the appropriate directories (i.e., in subdirectories corresponding to the `SYSTEM` and `SYSTEM_CPU` environment variables) below `/usr/local/wind/wind`.

## 9 Parallel Processing

The time required to compute a solution can be reduced by using the distributed parallel processing capability of WIND. WIND can simultaneously use multiple systems connected via a network as though they were a single computer. The systems are typically workstation class machines and need not be all of the same vendor.

When operating in distributed parallel processing mode, the system on which the job originates is called the *master* system. All other systems are called *worker* systems. WIND distributes complete zones to participating worker systems. Each zone is solved in parallel with other zones on other systems. The systems exchange boundary information during each cycle in order to propagate information throughout the computational domain. There may be fewer systems than zones to be computed, in which case, a system will be assigned another zone when it finishes its current assignment. The user specifies the names of the participating worker systems via the *multi-processing control file*, described in [Section 9.1](#).

If a worker system fails due to either a system or network failure during the course of a run, the job will restart from the last checkpoint (see the **checkpoint** directive in [Section 9.1](#)) without the failed system. The automatic restart ability will be invoked as many times as necessary during a job until no more systems are available.

### 9.1 Multi-Processing Control File

The multi-processing control file specifies the hosts that will be available as well as some miscellaneous options. The presence of a valid multi-processing control file is all that is required to utilize the distributed parallel processing capability of WIND. If the main WIND input data file name is *input.dat*, the name of the multi-processing control file must be *input.mpc*. When this file is present, the *wind* script will ask the user if they really want to use multi-processing mode.

Comments may be included in the file with the normal WIND comment indicator *'/'*, or additionally *'#'*. Blank lines are ignored. Trailing comments are not allowed. The formats of the directives follow.

```
host {localhost | hostname} [nproc n]
```

**host** directives specify the names of the worker systems (given by the *hostname* parameter) that will be used to process zones. In general, there should be one **host** directive for each worker system. If a particular system appears more than once, each occurrence is treated as a unique system and will process assigned zones simultaneously. This is not advisable unless the system has multiple processors and sufficient memory.

The optional parameter **nproc** *n* may be used to specify the number of processes to allow to run in parallel on the specified host. It is equivalent to repeating the **host** directive *n* times.

If no **host** entries appear in the multi-processing control file, the originating system will automatically be selected as the only host. When used on a system with sufficient memory and the **assignment mode dedicated** directive, the normal I/O associated with a single processor solution will be eliminated (except for checkpoints).

The special parameter **localhost** is used when running on a single multi-processing system and the system name is unknown at the time of job submittal, such as for batch systems (like NQE) that

can spawn to multiple systems or clusters of servers. Using `localhost` is preferred over not putting in any `host` directives because it assures that the scripts set up WIND consistently.

`host` entries should appear in the file in decreasing order of computational power. WIND assigns the most computationally intensive zones to the highest entries in the list.

The system that originates the job is not automatically included in the host list. If it is desired to also assign solution tasks to the originating system, it should have a host entry like any other system. For estimating purposes, the master process typically consumes less than one percent of the CPU time on the master host.

```
i/o {direct | indirect}
```

This directive specifies the type of access that worker systems have to files on the master. The default is `indirect`, which means that workers do not have access to the files on the master, and that file I/O must therefore be done using message passing to/from the master process.

On multi-processor systems, however, `i/o direct` may be used to indicate that the worker processes may access the files directly, bypassing communication through the master process. This significantly reduces communication overhead and increases performance by as much as 10-40%.

There are a couple of things to be aware of when using the `i/o direct` option. First, it should only be used when running on a single multi-processor system, not with workstation clusters. And second, the number of open files per process must be set large enough.

```
checkpoint {[every] {time minutes | count cycles} | none}
```

This directive specifies how often the worker systems transfer their flow field information to the flow file on the master system. In the event of a failure, the solution is automatically restarted from the last checkpoint. Specifying too small a number can result in very high network overhead and low throughput. A large number improves performance but can cause wasted time if a lot of network failures occur. If `checkpoint none` is specified, the flow field information is updated only at the end of the job. The default value is:

```
checkpoint every 60 minutes
```

```
assignment mode {dedicated | shared | transient}
```

`assignment mode` controls how tasks are assigned to target systems. There may be multiple appearances of this directive. They affect subsequent `host` entries up to the next `assignment mode` directive. A description of each mode follows:

**dedicated** Each task (zone) gets a unique Unix process on the target system. If a system must process more than one zone, each will have a separate process, but only one will be allowed to run at a time unless multiple `host` entries are present for the system. This is the default mode and should not be changed unless there is insufficient swap space for the processes assigned to the host.

- shared** Unless a system must process more than one zone, this mode is the same as **dedicated**. If more than one zone must be processed, only one Unix process is allocated and individual zones are swapped to and from local disk on the target system. This mode should be used only if the target system does not have sufficient swap space to contain the zones it needs to process.
- transient** When a task completes, it writes all of its flow information back to the master processor and terminates. A new Unix process is then started for the next cycle. This mode is for future use and should not be specified.

```
route [indirect | direct]
```

Controls how data is routed between the master and worker tasks. This value should not be changed unless directed by NPARC-support.

- indirect** All messages go from the task on the local machine to the local PVM daemon, over the network to the remote PVM daemon, which forwards it to the remote task.
- direct** All messages go directly from task to task, bypassing the PVM daemons. This mechanism, while supposedly more efficient, cannot be used when the number of zones is large.

```
#LOADLIMIT limit
```

The WIND/PVM initialization script will automatically eliminate workers that are deemed “too busy.” A system is defined to be “too busy” when its 15 minute load factor, as reported by the Unix “uptime” command (the last number on the line) is greater than a certain limit (0.60 by default). The purpose of this check is to eliminate hosts that are occupied by hung processes.<sup>20</sup>

The load factor for each worker will be displayed in the list output file at the top with the other messages that occur during the preparation of the workers. The load factor will be displayed as a percent (0.60 corresponds to 60%). Note that the load factors in excess of 100% are possible. A message will also be displayed if the load factor exceeds the allowed threshold.

Occasionally, there is a problem with the “uptime” command and it reports a high load factor when there is no load on the system. To avoid this problem, the **#LOADLIMIT** directive<sup>21</sup> may be used to override the default value of 0.60. The parameter *limit* specifies the load limit for all hosts up to the next **#LOADLIMIT** directive. A **#LOADLIMIT** directive with no parameters restores the default load limit. This command should only be used when you know that including an overloaded host will not affect your job.

The following example illustrates the use of the **#LOADLIMIT** directive in the multi-processing control file.

<sup>20</sup>Note that the load factor is checked only at initialization time. If the system becomes busy during a run ... well ... too bad.

<sup>21</sup>Note that this directive is an exception to the use of **#** as a comment indicator.

```
# Next statement considers hosts ws1463 and ws1464 loaded
#   only if their load factor exceeds 100%
#loadlimit 100
host ws1463
host ws1464
# The next statement restores the default load limit
#loadlimit
host ws1465
# Use a really high limit for ws1466 - disables the limit check
#loadlimit 9999
host ws1456
```

Another way to modify the default load limit is to set the `PVM_LOAD_LIMIT` environment variable *before* you submit your job. For example, using the Bourne shell:

```
$ PVM_LOAD_LIMIT=75
$ export PVM_LOAD_LIMIT
$ wind
```

## 9.2 Multi-Processing Script Considerations

### Unix Security Considerations

The network communication software uses remote shell commands to start the worker tasks. This implies the user must have a valid account on the worker system. In addition, the user name on the master system and all worker systems must be the same.<sup>22</sup>

To allow remote shell commands to access a worker system, the host name of the master system must be in the file `.rhosts` in the user's home directory on the worker system, or in the system file `hosts.equiv` on the worker system. [If `ssh` is being used, the corresponding file names are `.shosts` and `shosts.equiv`. These files are used in exactly the same way as `.rhosts` and `hosts.equiv` and on a non-`ssh` system.] Note that this is required even if the master and worker are the same system.

The `.rhosts` file is a text file containing a list of system names, and the userids on each of those systems, that are allowed to access the current host via a remote shell command. The file should have its permissions set to `rw----`, so issue the following command after creating the file:

```
chmod 600 .rhosts
```

Once the `.rhosts` file has been created, it may be tested by issuing the following command from the system where the job will be submitted.<sup>23</sup>

```
rsh worker-name ls -la
```

Things are functioning properly if the directory listing appears. In many situations the users home directory is located on a file server, so only one `.rhosts` file needs to be maintained.

More information about `.rhosts` files may be found by entering `man rhosts` on most Unix systems.

---

<sup>22</sup> By "remote shell commands" we mean `rsh` (`remsh` on HP and Cray systems) and `rcp`; or, if the `ssh` (secure shell) software is used, `ssh` and `scp`. Note that if `ssh` is to be used, that must be specified during the creation of the PVM and WIND executables. See the *WIND Installation Guide* for more information.

<sup>23</sup> Use the command `remsh` or `ssh` instead of `rsh`, if that's appropriate for your system.



Ensuring PVM Jobs Stop at a Specific Time

To avoid the possibility of a PVM job continuing outside an allotted off-shift window, a series of scripts can be executed by the Unix cron process. In the WIND distribution, these scripts are in the directory *wind/cfd/bin/pvmkill*. Four files are located there:

1. *cronkill* — This file tells the continuous running job scheduler when to terminate processes. The first two digits on each line are the minute, the third digit is the hour, and following the \*'s are the days when each of the commands will be executed (Monday = 1). The first command is the 'nicest' way to kill the job, with the following two successively harsher. Note that this file must be edited so that output goes to your directory and the paths for the scripts are correct.
2. *pvmclean* — A script which terminates jobs in a relatively nice fashion.
3. *naskill* — A script which terminates jobs in a bit harsher fashion.
4. *naspvkill* — A script which terminates jobs in the meanest fashion.

To invoke these processes, copy the above scripts to *each* master you're using, edit *cronkill* appropriately, and insert these processes into the crontab on *each* master by entering

```
crontab cronkill
```

To check if this worked, enter

```
crontab -l
```

which will give a list of all your cron entries.

### 9.3 Multiple Parallel Jobs

When running in parallel mode on a cluster of workstations, the master system and all worker systems being used by a given user cannot be used by any other parallel job from the same user as long as the first job is active. A different user, however, can have a parallel job running simultaneously on the same systems, assuming that the memory, disk space, etc., are sufficient to support multiple jobs.

There are no restrictions on the number of parallel jobs for a given user on a multi-processor system (i.e., using the *-mp* option to the *wind* script; see [Section 8.1](#)), again assuming that the computer resources are available to support multiple jobs.

### 9.4 Hints

Because synchronization takes place at the end of each cycle, total throughput is established by the processor that takes the longest to complete its assigned work. The optimum situation is to have all zones of equal size and have one processor for each zone. This gives maximum throughput and processor utilization, but is generally not achievable. If all zones cannot be close to the same size, a mixture of sizes is preferable. The case to avoid is a configuration with one zone of comparable size to the sum of the remaining zones. In this case, one can achieve at most a factor-of-two performance improvement regardless of the number of processors used. In general, if *n* is the number of points

in the largest zone and  $N$  is the total number of points, the maximum possible speed up is  $N/n$  (assuming identical processors and similar algorithm specification).

Given a number of processors  $P$  with relative speeds  $p_i$  (larger  $p$  implies faster), and a number of zones  $N$  of sizes  $n_j$ , the assignment of work is done as follows:

1. Assign the largest zone  $j$  to processor 1 and compute  $T_1 = n_j/p_1$ .
2. Repeat step 1 for the remaining  $P - 1$  processors, assigning the largest remaining zone  $j$  to processor  $i$  and compute  $T_i = n_j/p_i$ .
3. If any zones remain to be assigned, locate processor  $i$  such that  $T_i$  is a minimum. Assign the largest remaining zone  $j$  to processor  $i$ , computing  $T_i = T_i + n_j/p_i$ .
4. Repeat step 3 for remaining unassigned zones.

Consider adding processors if  $T$  for any processor is significantly larger than the others, and that processor has more than one zone assigned.

The output file from a run will indicate what zones are assigned to what processor, and will have a report containing the utilization of each processor.

## 10 Keyword Reference

### 10.1 Text Conventions

In this manual, keywords are indicated by upper-case words in a fixed-pitch font, **LIKE THIS**. In the actual input file, however, they may be entered as upper or lower case. They may also be abbreviated. WIND checks for keywords in quasi-alphabetical order and does *not* check for uniqueness. Therefore, specifying “A I” for arbitrary inflow is not a good idea. In cases where multiple options are available, the default (if one exists) is underlined.

User-specified input parameters are indicated by italics, *like this*. These user-specified parameters are usually numeric values, but may also be other keywords.

Most keywords consist of a single line, containing the keyword and its user-specified input parameters. Other keywords, like **CHEMISTRY**, indicate the start of a keyword block containing several lines, bracketed by starting and ending keywords. Within a keyword block, only individual keywords relevant to that block may be used. Keyword blocks are indicated by the word “block” in parentheses. Some keywords have several options, making them long and complex. These are sometimes split into two (or more) lines for display purposes, with a “\” at the end of the line being continued, but must nevertheless be on a single line in the input data file.

The various keywords and keyword blocks are listed alphabetically, with each one starting a new page. The overall syntax for each keyword or keyword block is shown in a box at the top of the page, with the details following.

In addition, the following documentation conventions are also used:

- | The “or” symbol; used to separate multiple choices
- [] Delimiters surrounding optional entry(s)
- { } Delimiters surrounding multiple entries when exactly one of them is required

### 10.2 Zone Selection

Many of WIND’s capabilities may be specified on a zone-by-zone basis. Keywords used to enable these capabilities may include a zone selector at the end of the keyword command in the input data file. The keywords for which this type of specification is valid include an optional *zone\_selector* format specifier in their description. The *zone\_selector* must be of the following form:

[ZONE] *zone-list*

where *zone-list* is of the form:

*range1* [, *range2* [, ... *rangen*]]

A *range* is of the form:

|                                 |                                                                                                      |
|---------------------------------|------------------------------------------------------------------------------------------------------|
| <i>zonenum</i>                  | Selects zone <i>zonenum</i>                                                                          |
| <i>begzone</i> : <i>endzone</i> | Selects all zones from <i>begzone</i> to <i>endzone</i>                                              |
| <i>begzone</i> :                | Selects all zones from <i>begzone</i> to <b>MAXZONE</b> , the total number of zones in the grid file |

|                       |                                            |
|-----------------------|--------------------------------------------|
| <code>:endzone</code> | Selects all zones from 1 to <i>endzone</i> |
| <code>ALL</code>      | Selects all zones                          |

For keywords that turn on capabilities that may be selected by zone, omitting the *zone\_selector* defaults to using the capability in all zones. You should therefore change the default selection before changing individual zones, as changing the default will reset any zones which have been individually set earlier in the input file.

### Example

To turn off (i.e., perform no iterations) in zones 5, 6, and 7, but still pass the information in these zones to the adjacent zones, and to use 10 iterations per cycle in all other zones, use the following sequence of keywords:

```
ITERATIONS PER CYCLE 10
ITERATIONS PER CYCLE -1 ZONE 5:7
```

Note that reversing the order of the keywords would not work because resetting the default will override the initial selection.

## 10.3 Keyword Details

The following is an alphabetical list of all the WIND keywords, with detailed descriptions of their syntax and usage. Every attempt was made to build keyword inputs from intuitive, English-language words, and to adhere to some general rules of construction for the keyword commands in the data file.

**ACCELERATE** — Convergence acceleration (block)

```
ACCELERATE
  ZONE n
    [SECOND START iter1 END iter2]
    [FOURTH START iter1 END iter2]
    [CFLRAMP START iter1 END iter2] {nzn1 nzn2 | 0}
ENDACCELERATE
```

The objective of WIND's convergence acceleration scheme is to approach the steady state solution more quickly by allowing the use of a large CFL number early in the calculation, adding second- and fourth- order smoothing to maintain numerical stability. This option must be used in conjunction with the [SMOOTHING](#) keyword and [TEST 49](#).

*iter*<sub>1</sub> and *iter*<sub>2</sub> specify the starting and ending iteration values for the corresponding keyword parameter. The CFL number will be ramped down from its starting value, specified using the [CFL#](#) keyword, to 1.0 using the formula

$$\text{CFL}_i = 1 + (\text{CFL}_{i-1} - 1) \frac{i - \text{iter}_2}{\text{iter}_1 - \text{iter}_2}$$

where *i* is the current iteration number.

If starting and ending iteration values are not specified, the following default values are used:

| <u>Option</u> | <u><i>iter</i><sub>1</sub></u> | <u><i>iter</i><sub>2</sub></u> |
|---------------|--------------------------------|--------------------------------|
| SECOND        | 1                              | 150                            |
| FOURTH        | 50                             | 500                            |
| CFLRAMP       | 50                             | 500                            |

#### Example

In the following example, convergence acceleration is applied in zones 3 and 4. Second-order smoothing is applied between iterations 1 and 200, and fourth-order smoothing is applied between the default values of iterations 50 and 500. The CFL number specified with the [CFL#](#) keyword will be used for the first 250 iterations, then ramped down to a value of 1.0 between iterations 250 and 500.

```
SMOOTHING SECOND s1 FOURTH s2 SMLIMIT 0.
ACCELERATE
  ZONE 3
  ZONE 4
    SECOND START 1 END 200
    CFLRAMP START 250 END 500
ENDACCELERATE
TEST 49 2
```

Recommended values for the smoothing parameters *s*<sub>1</sub>, *s*<sub>2</sub>, and for the starting CFL number, are:

| <u>Dimensions</u> | <u><i>s</i><sub>1</sub></u> | <u><i>s</i><sub>2</sub></u> | <u>CFL #</u> |
|-------------------|-----------------------------|-----------------------------|--------------|
| 2D                | 0.1                         | 0.03                        | 10–15        |
| 3D                | 0.06                        | 0.01                        | 2.5          |

See Also: [SMOOTHING](#), [CFL#](#), [TEST 49](#)

**ACTUATOR | SCREEN** — Discontinuous change across a zone boundary (block)

```
{ACTUATOR | SCREEN}
  ZONE iz1 BOUNDARY {I1 | IMAX | J1 | JMAX | K1 | KMAX} \
    [SUBSET I range J range K range]
  ZONE iz2 BOUNDARY {I1 | IMAX | J1 | JMAX | K1 | KMAX} \
    [SUBSET I range J range K range]
  TURNING {CONSERVE {ANGLE | PARALLELU} | \
    ZERO PARALLELU | \
    {SOLIDBODY | VORTEX} val xc yc zc | \
    SPECIFY ANGLE  $\alpha$  [ROTATE  $\beta$ ]}
  TIP-EFFECT r1 r2 r3 r4
  POWER {{DPS | DPT | DPOWER} val | \
    TURNING | \
    {SOLIDBODY | VORTEX} val xc yc zc}
  EFFICIENCY {ETA val | \
    CLOSS val | \
    VORTEX val | \
    SCREEN {NORMAL | TOTAL} SOLIDITY sol}
{ENDACTUATOR | ENDScreen}
```

This keyword enables the user to specify a discontinuous change in properties across a zone boundary or portion of a zone boundary. Examples include actuator disks, engine face models, and screens.

The following restrictions apply:

- The **BOUNDARY TVD FACTOR 0** keyword option should be used for all actuator disk and screen boundaries.
- Screens require zero work (POWER DPOWER 0.)
- Only one storage location for center of rotation, the last one encountered is used for all centers.

The various elements of the ACTUATOR | SCREEN input block are defined as follows:

```
{ACTUATOR | SCREEN}
```

Defines the beginning of the actuator or screen block.

```
ZONE iz1 BOUNDARY {I1 | IMAX | J1 | JMAX | K1 | KMAX} \
  [SUBSET I range J range K range]
ZONE iz2 BOUNDARY {I1 | IMAX | J1 | JMAX | K1 | KMAX} \
  [SUBSET I range J range K range]
```

These two lines define the location of the actuator disk or screen. The relevant zones are given by the values of *iz1* and *iz2*, and the relevant boundaries within zones *iz1* and *iz2* are specified via the BOUNDARY keyword parameter.

*iz1*     Zone to which increments will be added when passing info to *iz2*

*iz2* Zone receiving positive increments, increments will be subtracted when passing information back to zone *iz1*

The **SUBSET** parameter may be used to specify that the change in properties occurs only over a part of the zone boundary. Otherwise, it is assumed that the change occurs over the entire boundary. The *range* parameters define the part of the zone boundary over which the change occurs, and take one of the following forms:

*index1 index2* Starting and ending indices in the specified direction. **LAST** may be used for the last index.

**ALL** Equivalent to 1 **LAST**.

The starting and ending indices for the appropriate **I**, **J**, or **K** parameter (depending on the boundary specified) must be the same, and correspond to that boundary. In addition, for two-dimensional cases, the **K** parameter must be specified as either **K 1 1** or **K ALL**.

```
TURNING {CONSERVE {ANGLE | PARALLELU} | \
        ZERO PARALLELU | \
        {SOLIDBODY | VORTEX} val xc yc zc | \
        SPECIFY ANGLE α [ROTATE β]}
```

Defines the net change in parallel velocity across the zone boundary.

**CONSERVE** Conserves the net flow angle (**ANGLE**) or the parallel velocity components (**PARALLELU**) across the zone boundary. (The **ANGLE** option is currently not implemented.)

**ZERO PARALLELU** Sets the parallel components of velocity across the zone boundary to zero

**SOLIDBODY** Defines a solidbody rotation increment to the parallel velocity, where:

*val* Rotation rate in radians/second (positive by right hand rule)

*x<sub>c</sub>, y<sub>c</sub>, z<sub>c</sub>* Center of rotation (must be in the plane, requires the zone boundary to lie in a constant *x*, *y*, or *z* plane) (inches)

**VORTEX** Defines free vortex flow increment to parallel velocity, where:

*val* Vortex strength  $\kappa$  (ft<sup>2</sup>/sec), where  $\kappa = \omega a^2 = \Gamma/2\pi$ , and  $\Gamma$  is the circulation,  $\omega$  is the rotation rate of the solidbody core, and  $a$  is the radius of the solidbody core (required to avoid  $P = 0$  at axis),  $a^2 = \kappa^2 \rho / (0.9 P_\infty)$  (assumes  $P_{min} = 0.1 P_\infty$ )

*x<sub>c</sub>, y<sub>c</sub>, z<sub>c</sub>* Center of rotation (inches)

**SPECIFY ANGLE** Allows the user to specify the flow turning angle.

*alpha* The flow angle giving the rotation of the *iz2* boundary normal, projected onto the *x-y* plane, about the *z*-axis (degrees)

*beta* An optional rotation of the resulting vector about the *y*-axis

**TIP-EFFECT** *r1 r2 r3 r4*

Forces increments to go to zero at hub and/or tip to avoid solution discontinuities at the boundaries. A scalar, (0–1) multiplies the turning and power when this option is on. This is required for engine face models (where the wall velocity at the tip must be zero in the diffuser frame of reference). *r1–r4* define linear regions ranging from 0 to 1 between *r1* and *r2*, and from 1 to 0 between *r3* and *r4*. *r1*, *r2*, *r3*, and *r4* define the distance from the center of rotation (inches).

This keyword requires that **TURNING SOLIDBODY**, **TURNING VORTEX**, or **POWER SOLIDBODY** be specified.

**POWER** **{DPS | DPT | DPOWER}** *val* | \
  
**TURNING** | \
  
**{SOLIDBODY | VORTEX}** *val x<sub>c</sub> y<sub>c</sub> z<sub>c</sub>*

Defines the power increment across the zone boundary. Screens require setting the power to zero. i.e., **POWER DPOWER 0**.

**DPS** *val* specifies the static pressure increment across the actuator boundary (psi). Requires that the efficiency be specified, using **EFFICIENCY ETA**.

**DPT** *val* specifies the total pressure increment across the actuator boundary (psi). Requires that the efficiency be specified, using **EFFICIENCY ETA**.

**DPOWER** *val* specifies a (constant) power per unit area increment. (ft-lb/sec-ft<sup>2</sup>). (Corresponds to unsteady (rotor) free vortex turning)

$$\text{DPOWER} = \rho u c_p (T_{t2} - T_{t1})$$

**TURNING** Specifies work corresponding to the net turn across the zone boundary (specified in the **TURNING** element). Assumes all turning is done in an unsteady process (by the rotor), i.e., no stator.

$$dW = c_p (T_{t2} - T_{t1}) = \omega r (w_2 - w_1)$$

where  $\omega$  is the rotation rate of the rotor,  $r$  is the local radius from the center of the rotor, and  $w$  is the local circumferential velocity.

This option requires that **TURNING SOLIDBODY** be specified.

**SOLIDBODY | VORTEX** Defined as in the **TURNING** element. This defines the turning accomplished by the rotor. The net turning may be altered by another process (e.g., by a stator).

$$dW = c_p (T_{t2} - T_{t1}) = \omega r (w_2 - w_1)$$

Note: Currently vortex turning (i.e., **POWER VORTEX**) is not allowed. This would correspond to constant work across the rotor. However, currently, the procedure used to eliminate the vacuum at the core (setting  $P_{min} = 0.1P_\infty$ ) makes the work input independent of the strength of the vortex, so the user could not vary the work input by changing  $\kappa$ .

**EFFICIENCY** **{ETA** *val* | \
  
**CLOSS** *val* | \
  
**VORTEX** *val* | \
  
**SCREEN** **{NORMAL | TOTAL}** **SOLIDITY** *sol*



Defines the efficiency of the actuator disk or screen.

- ETA** Compressor efficiency,  $val = [(P_{t2}/P_{t1})^{(\gamma-1)/\gamma} - 1]/[(T_{t2}/T_{t1}) - 1]$
- CLOSS** Loss coefficient,  $val = (P_{t1} - P_{t2})/q$ , where  $q = \rho U^2/2$  ( $U$  based on normal Mach number)
- VORTEX** Free vortex total pressure loss. The value  $val$  is the maximum value of  $(P_{t2} - P_{t1})/P_{tinf}$  (i.e., the loss at the center of the vortex). A linear distribution is assumed from the vortex center to the radius  $a$ , where  $a$  is determined by the strength value specified using **TURNING VORTEX**, or directly using [TEST 180](#).  
This option requires that **TURNING VORTEX** be specified.
- SCREEN** Use screen loss relations to define total pressure loss, where
- NORMAL** Use normal component of Mach number
  - TOTAL** Use total Mach number. (This option is not currently implemented.)
  - sol* Solidity of screen =  $A_b/(A_b + A_o)$ , where  $A_b$  is the blocked area, and  $A_o$  is the open area.

If solidity is specified, the screen loss coefficient associated with the screen model is defined by the solidity correlation of [Cornell \(1958\)](#), unless the optional **CLOSS** value is specified.

The screen model is not intended for use with choked screens, where the screen is significantly limiting the mass flow rate. During the solution start-up phase, it may be necessary to specify a low solidity, then increase it to the desired value to avoid strong choking in transients.

This option requires that the power be zero. i.e., **POWER DPOWER 0**.

{ENDACTUATOR | ENDScreen}

Ends actuator or screen input block

### Examples

The following examples illustrate the use of the **ACTUATOR | SCREEN** input block for an engine face and for a screen.

#### *Engine face model*

```

ACTUATOR
  ZONE 1 BOUNDARY IMAX
  ZONE 2 BOUNDARY I1
  TURNING SOLIDBODY 240000. 312. 54. 0.
  TIP-EFFECT 5. 5.1 39.8 40.0
  POWER TURNING
  EFFICIENCY ETA 0.85
ENDACTUATOR
BOUNDARY TVD FACTOR 0 ZONE 1 BOUNDARY IMAX
BOUNDARY TVD FACTOR 0 ZONE 2 BOUNDARY I1

```

#### *Screen*

```
SCREEN
  ZONE 3 BOUNDARY K1
  ZONE 2 BOUNDARY IMAX
  TURNING ZERO PARALLELU
  POWER DPOWER 0.0
  EFFICIENCY SCREEN NORMAL SOLIDITY 0.1
ENDSCREEN
BOUNDARY TVD FACTOR 0 ZONE 3 BOUNDARY K1
BOUNDARY TVD FACTOR 0 ZONE 2 BOUNDARY IMAX
```

*See Also:* [BOUNDARY TVD](#), [TEST 180](#)

**ARBITRARY INFLOW** — Arbitrary inflow (block)

```

{ARBITRARY INFLOW | DIFFUSER INFLOW}
  [STATIC | TOTAL]
  [HOLD_TOTALS | HOLD_CHARACTERISTICS]
  ZONE n
  [UNIFORM [M P T α β] [valk valom]]
    [sp1 sp2 ... spn]]
  [IJK_RANGE [FROZEN] imin imax jmin jmax kmin kmax M P T α β [valk valom]]
    [sp1 sp2 ... spn]]
  [UNSTEADY var_name freq ampl phase]
  [{VORTEX | SOLIDBODY | ROTATESOLID} Mn P T α β xc yc zc
    {dw1 | dw1 dw2 dw3}]
  [USERSPEC fs bl1 bl2 npts
    y1 M P T α β
    y2 M P T α β
    ...
    yn M P T α β]
  [USERCHEM fs bl1 bl2 npts
    y1 M P T α β sp1 sp2 ... spn
    y2 M P T α β sp1 sp2 ... spn
    ...
    yn M P T α β sp1 sp2 ... spn]
  [USERKE fs bl1 bl2 npts
    y1 M P T α β valk [valom]
    y2 M P T α β valk [valom]
    ...
    yn M P T α β valk [valom]]
  [USERCHEMKE fs bl1 bl2 npts
    y1 M P T α β sp1 sp2 ... spn valk [valom]
    y2 M P T α β sp1 sp2 ... spn valk [valom]
    ...
    yn M P T α β sp1 sp2 ... spn valk [valom]]
[ENDINFLOW]

```

Several options are available to set conditions at arbitrary inflow boundaries. The default is uniform inflow (i.e., no boundary layer) at the conditions that are set using the [FREESTREAM](#) keyword. Other options are selected by using either of the equivalent keywords **ARBITRARY INFLOW** or **DIFFUSER INFLOW**.

The remaining lines select the specific type of nonuniform entrance data to be provided. Discussion of the input data for each of these options is presented in the remainder of this subsection, grouped by Control Functions, Condition Specification, and Special Capabilities. These keywords can start in any column. Generally, they should be indented from the **ARBITRARY INFLOW** keyword to set them apart.

This keyword may also be used to initialize (or reinitialize) the flow conditions within the specified zone, as described in [Section 3.5](#).

Note: The [GAS](#) keyword and the [CHEMISTRY](#) keyword block, if used, must come before the **ARBITRARY INFLOW** keyword block in the input data (*.dat*) file.

## Control Functions

ENDINFLOW

This optional keyword may be used to end the arbitrary inflow definition.

STATIC | TOTAL

Arbitrary inflow conditions specified after this keyword will be considered as static or total, depending which of these is set. By default, the flag is set to the value from the [FREESTREAM](#) keyword.

Note: TOTAL is only valid for an ideal gas.

HOLD\_TOTALS | HOLD\_CHARACTERISTICS

Specifying HOLD\_TOTALS indicates that total temperature and local flow angles are to be held at their specified values. The conditions are held fixed for all the arbitrary inflow regions in the specified zone. This option only works in conjunction with the UNIFORM or IJK\_RANGE keyword options.

The default is HOLD\_CHARACTERISTICS, which indicates that characteristic values are to be held constant.

Note that the total pressure is always held fixed, whether HOLD\_TOTALS is specified or not. Note also that although the Mach number is specified with the UNIFORM and IJK\_RANGE keyword parameters, it may be adjusted during the boundary condition treatment.

The specified option will remain in effect for all following ZONE keywords.

ZONE *n*

This keyword, which *must* be specified, identifies the zone for which inflow conditions are being set. For example, if zone 2 is an internal jet, conditions other than freestream may be desired at the inflow to zone 2.

Within the ARBITRARY INFLOW keyword block, the ZONE keyword must come before the keyword defining the inflow conditions for that zone. In addition, inflow conditions may only be specified for one zone at a time.

Example

The following ARBITRARY INFLOW block specifies that total conditions are to be held constant at arbitrary inflow surfaces in zones 1 and 2, with  $M = 3.5$ ,  $P_T = 251.15$  psi, and  $T_T = 1167.9$  °R. In zone 3, characteristic values are to be held constant at arbitrary inflow surfaces, consistent with the flow conditions given with the [FREESTREAM](#) keyword.

```

ARBITRARY INFLOW
  TOTAL
  HOLD_TOTALS
  ZONE 1
    UNIFORM 2.5 251.15 1167.9 0.0 0.0
  ZONE 2
    UNIFORM 2.5 251.15 1167.9 0.0 0.0
  HOLD_CHARACTERISTICS

```

ZONE 3  
UNIFORM  
ENDINFLOW

### Condition Specification

UNIFORM [*M P T α β*] [*valk valom*]]  
[*sp1 sp2 ... spn*]

The entrance flow is uniform, at the conditions specified. If no flow conditions are specified, those specified with the **FREESTREAM** keyword are used. No other data is provided to select this option. This is the default option, and is selected automatically if no extended input options are selected.

Different conditions can be specified within each zone. This is done by adding the run conditions after the keyword. These data are added in free format, one or more blanks separating values.

|                            |                          |
|----------------------------|--------------------------|
| <i>M</i>                   | Mach number              |
| <i>P</i>                   | Pressure                 |
| <i>T</i>                   | Temperature              |
| <i>α, β</i>                | Angles of attack and yaw |
| <i>sp1, sp2, ... , spn</i> | Species mass fractions   |

Pressure and temperature are static or total, depending on whether **STATIC** or **TOTAL** is specified. As always, pressure is specified in pounds per square inch, temperature in degrees Rankine, and *α* and *β* in degrees.

If the SST turbulence model is being used (see the **TURBULENCE** keyword), *valk* and *valom* may be used to specify inflow turbulence levels. Note that you may specify either *valk*, or *valk* and *valom*, but not *valom* by itself. Note also that if these values are being specified, the **TURBULENCE** keyword must come before the **ARBITRARY INFLOW** keyword block in the input data (*.dat*) file.

The following options are possible:

*valk* > 0      The turbulent kinetic energy *k* and the specific dissipation rate *ω* are specified directly, with

$$k = valk \text{ (ft}^2\text{/sec}^2\text{)}$$

$$\omega = valom \text{ (1/sec)}$$

The turbulent viscosity  $\nu_t$  is then equal to  $k/\omega$ .

*valk* < 0      The turbulent kinetic energy *k* is set equal to *valk* percent of the “reference” kinetic energy  $U^2/2$ , where *U* is computed from the specified values of *M* and *T*. Thus

$$k = 0.01 |valk| \frac{U^2}{2}$$

The turbulent viscosity  $\nu_t$  is automatically set equal to  $0.001\nu_l$ , where  $\nu_l$  is the laminar viscosity, and the specific dissipation rate is computed as  $\omega = k/\nu_t$ .

$valom < 0$  The specific dissipation rate  $\omega$  is set equal to  $valom$  percent of  $U/L_{ref}$ , where  $U$  is computed from the specified values of  $M$  and  $T$ , and  $L_{ref}$  is the reference length from the grid (.cgd) file. Thus

$$\omega = 0.01 |valom| \frac{U}{L_{ref}}$$

The turbulent viscosity  $\nu_t$  is set to the same percentage of the laminar viscosity.

$$\nu_t = 0.01 |valom| \nu_l$$

The turbulent kinetic energy is then computed as  $k = \omega \nu_t$ .

If inflow turbulence levels are not specified using one of the above options, or if  $valk = 0$ , default values are computed from

$$\begin{aligned}\omega &= 10U/L_{ref} \\ \nu_t &= 0.001\nu_l \\ k &= \omega\nu_t\end{aligned}$$

Note that

- If  $valk > 0$ , a positive value must be specified for  $valom$ .
- If  $valk \leq 0$ ,  $valom$  should not be specified.
- If  $valom < 0$ , a value must also be specified for  $valk$ , but it is ignored.

#### Example

```
ARBITRARY INFLOW
UNIFORM 1.1 100. 900. 10. 0.
ENDINFLOW
```

```
IJK_RANGE [FROZEN] imin imax jmin jmax kmin kmax M P T  $\alpha$   $\beta$  [valk [valom]]
[sp1 sp2 ... spn]]
```

This keyword allows specification of inflow conditions over an arbitrary range of  $i$ ,  $j$ , and  $k$  indices on any computational boundary plane. The user specifies the minimum and maximum  $i$ ,  $j$ , and  $k$  indices which describe the region, followed by the flow conditions to be applied, as follows:

|                 |                                                     |
|-----------------|-----------------------------------------------------|
| $imin, imax$    | Minimum and maximum $i$ indices bounding the region |
| $jmin, jmax$    | Minimum and maximum $j$ indices bounding the region |
| $kmin, kmax$    | Minimum and maximum $k$ indices bounding the region |
| $M$             | Mach number                                         |
| $P$             | Pressure                                            |
| $T$             | Temperature                                         |
| $\alpha, \beta$ | Angles of attack and yaw                            |

$sp_1, sp_2, \dots, sp_n$  Species mass fractions

Pressure and temperature are static or total, depending on whether **STATIC** or **TOTAL** is specified. As always, pressure is specified in pounds per square inch, temperature in degrees Rankine, and  $\alpha$  and  $\beta$  in degrees. There are no defaults for the index ranges.

The **FROZEN** option may be specified to freeze the inflow conditions over the indicated index range at their current values. Note that for a restart case (i.e., when a *.cfl* file already exists), the “current values” are those in the *.cfl* file, not those specified with the **IJK\_RANGE** parameter. For an initial run (i.e., when a *.cfl* file does not exist), the flow conditions will be frozen at the conditions specified with the **IJK\_RANGE** parameter.

If the SST turbulence model is being used (see the **TURBULENCE** keyword), *valk* and *valom* may be used to specify inflow turbulence levels. The various options are described above under the **UNIFORM** keyword.

Up to 500 **IJK\_RANGE** keywords may be specified. This is useful when specifying a boundary layer profile at an inflow boundary, or along solid walls during the flowfield initialization process, as described in [Section 3.5](#) starting on p. 35.

**UNSTEADY** *var\_name freq ampl phase*

This keyword allows the user to specify unsteady arbitrary inflow conditions, and it must be used with (and follow) the **IJK\_RANGE** keyword. Up to three different perturbations to the inflow conditions may be specified and will be superimposed to create unsteadiness centered about the conditions given with the **IJK\_RANGE** keyword.

*var\_name* One of the keywords **MACH**, **PRESSURE**, **TEMPERATURE**, **ALPHA**, **BETA**, or **VELOCITY**  
*freq* Frequency of the perturbation in Hertz  
*ampl* Amplitude of the perturbation in appropriate variable units  
*phase* Phase angle of the perturbation in degrees

Note that you may specify multiple, independent pairs of **IJK\_RANGE** and **UNSTEADY** keywords.

{**VORTEX** | **SOLIDBODY** | **ROTATESOLID**}  $M_n$   $P$   $T$   $\alpha$   $\beta$   $xc$   $yc$   $zc$  {*dw1* | *dw1 dw2 dw3*}

Specifies uniform inflow conditions with free-vortex or solid-body rotation superimposed. Solid-body rotation may be specified on any arbitrary inflow boundary. For free-vortex rotation, however, the arbitrary inflow boundary must be a constant  $x$ ,  $y$ , or  $z$  plane, and the center of rotation must lie on that plane.

$M_n$  Normal component of Mach number  
 $P$  Pressure (psia)  
 $T$  Temperature (°R)  
 $\alpha, \beta$  Average angles of attack and yaw  
 $xc, yc, zc$  Center of rotation in physical coordinates

For VORTEX,

$dw1$  Vortex strength. (See the ACTUATOR keyword.)

For SOLIDBODY and ROTATESOLID,

$dw1, dw2, dw3$   $x, y,$  and  $z$  components of the rotation rate vector (radians/sec)

The rotational velocity components are added in such a way that total pressure and total temperature are held constant at the inflow boundary. Thus, the TOTAL option should always be used for this mode, since with the STATIC option the computed static pressure and temperature at the inflow boundary may differ from the specified values.

For calculations in a rotating reference frame (see the ROTATE keyword):

- If the SOLIDBODY option is used, the total conditions specified are those in the rotating frame, and are held fixed in that frame. The total conditions in the inertial frame will vary.
- If the ROTATESOLID option is used, the total conditions specified are those in the inertial frame, and are held fixed in that frame. The total conditions in the rotating frame will vary. Note that the ROTATESOLID option is only valid for calculations in a rotating reference frame.

For both free-vortex and solid-body rotation, the flowfield must already be initialized. (See Section 3.5.) I.e., there must be a pre-existing .cfl file. The VORTEX, SOLIDBODY, and ROTATESOLID options cannot be used during a “cold” start.

## Special Capabilities

*The following arbitrary inflow keywords effect changes only to the  $i = 1$  computational plane.*

```

USERSPEC  fs  bl1  bl2  npts
          y1  M  P  T  α  β
          y2  M  P  T  α  β
          ...
          yn  M  P  T  α  β
    
```

This option allows the user to specify a 1-D profile normal to the surface, translated through some buttline range, below the vehicle. These conditions will be set last and thus the data will overwrite UNIFORM conditions over the range of interest.

|            |                                                                                           |
|------------|-------------------------------------------------------------------------------------------|
| $fs$       | Fuselage station of the profile (to be checked against the grid $i = 1$ fuselage station) |
| $bl1, bl2$ | Minimum and maximum buttline over which to translate the profile                          |
| $npts$     | Number of points defining the profile                                                     |
| $y1 - yn$  | Normal distance from the wall                                                             |
| $M$        | Mach number                                                                               |
| $P$        | Pressure (psia)                                                                           |
| $T$        | Temperature (°R)                                                                          |



$\alpha, \beta$  Angles of attack and yaw

The pressure and temperature may be the total or static conditions, depending upon the current setting of the **TOTAL/STATIC** keyword. If neither **STATIC** nor **TOTAL** have been specified under **ARBITRARY INFLOW**, then the existing switch from the global input parameters is used (default: **TOTAL**).

One profile can be specified for each zone. There can be 100 points in each profile. The normal distance is always assumed to be from  $j = 1$  (the reference wall is assumed to be at  $j = 1$ ). *bl1* is the minimum buttline and *bl2* is the maximum buttline.

See Also: [TEST 157](#)

By default, **USERSPEC** only specifies conditions below a vehicle. That is, the wall ( $j = 1$ ) must be above (higher  $y$ ) the interior grid points. **TEST 157** specifies that all points within the specified buttline range will be affected, above and below the vehicle. This should be the default, but isn't.

```
USERCHEM fs bl1 bl2 npts
  y1 M P T  $\alpha$   $\beta$  sp1 sp2 ... spn
  y2 M P T  $\alpha$   $\beta$  sp1 sp2 ... spn
  ...
  yn M P T  $\alpha$   $\beta$  sp1 sp2 ... spn
```

The **USERCHEM** option is identical to the **USERSPEC** option, except that chemistry species mass fractions  $sp_1, sp_2, \dots, sp_n$  are added to the end of each specified profile. Test options can then be set to model the mixing of gas streams which have different chemical compositions. At this time, only mixing can be modeled. The gas streams cannot chemically react.

Note: Only the **STATIC** input mode is available for chemistry.

Note: For [TEST 75 1](#), the *sp* input parameters specify the mass fractions of O<sub>2</sub>, CO<sub>2</sub>, H<sub>2</sub>O, NO<sub>2</sub>, and N<sub>2</sub>, in that order. For [TEST 75 2](#), they specify the mass fractions of O<sub>2</sub>, H, H<sub>2</sub>, H<sub>2</sub>O, and N<sub>2</sub>. The user must be careful to enter the mass fractions in the order shown, entering zeroes to leave out particular species.

Part of an example **USERCHEM** input block follows. The file sets up a rectangular jet where the jet composition is a mixture of O<sub>2</sub>, CO<sub>2</sub>, H<sub>2</sub>O, NO<sub>2</sub>, and N<sub>2</sub>. The last five columns hold the species mass fractions for the profiles. Note that the second-to-last column consists of zeroes; this is the unused position in the chemistry array which was described above for a four-species chemistry model.

```
ARBITRARY INFLOW
ZONE 1
USERCHEM 0.0 -10.0 10.0 6
  0.0  0.3  5.70  433.1  0.0  0.0  0.234  0.0  0.0  0.0  0.766
123.05 0.3  5.70  433.1  0.0  0.0  0.234  0.0  0.0  0.0  0.766
123.05 1.8  5.66 1940.0  0.0  0.0  0.096  0.120 0.048 0.0  0.736
133.75 1.8  5.66 1940.0  0.0  0.0  0.096  0.120 0.048 0.0  0.736
133.75 0.3  5.70  433.1  0.0  0.0  0.234  0.0  0.0  0.0  0.766
257.0  0.3  5.70  433.1  0.0  0.0  0.234  0.0  0.0  0.0  0.766
ENDINFLOW
TEST 157 1      USERSPEC ABOVE AND BELOW VEHICLE
```

```

USERKE fs bl1 bl2 npts
      y1 M P T α β valk [valom]
      y2 M P T α β valk [valom]
      ...
      yn M P T α β valk [valom]

```

The USERKE option is identical to the USERSPEC option, except that the values *valk* and *valom* are added to specify inflow turbulence levels when the SST turbulence model is being used. The various options are described above under the [UNIFORM](#) keyword.

```

USERCHEMKE fs bl1 bl2 npts
      y1 M P T α β sp1 sp2 ... spn valk [valom]
      y2 M P T α β sp1 sp2 ... spn valk [valom]
      ...
      yn M P T α β sp1 sp2 ... spn valk [valom]

```

The USERCHEMKE option is identical to the USERCHEM option, except that the values *valk* and *valom* are added to specify inflow turbulence levels when the SST turbulence model is being used. The various options are described above under the [UNIFORM](#) keyword.

See Also: [TURBULENCE](#), [REINITIALIZE](#), [ROTATE](#), [TEST 176](#)

**AXISYMMETRIC | AXI-SYM** — Axisymmetric flow

**{AXISYMMETRIC | AXI-SYM}** *yc theta*

This keyword allows a two dimensional grid to be run assuming axisymmetric flow. Note that  $k_{max}$  must be 1, and that the *.cgd* file is unaware that the grid is axisymmetric or 2D; that is determined at WIND run time.

The user-specified values are defined as:

- |              |                                                                                                                                                 |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>yc</i>    | Waterline location of rotation axis, in input units (scaled in WIND by <b>XLONG*SCAFAC</b> and offset by <b>YNOSE</b> )                         |
| <i>theta</i> | Degrees of rotation assumed for metrics. (Changing <i>theta</i> should not affect the solution, but will affect convergence. I like 5 degrees.) |

**BLEED** — Bleed region flow rate

```
BLEED {ibrg blv1 | POROSITY ibrg blv1 blv2 blv3 | MODEL ibrg mode blv1 blv2 blv3 blv4 | \
      FORCING ibrg blv1 blv2 blv3 | AEDC ibrg}
```

The effect of bleed on the flow can be modeled, if bleed regions were identified in the grid file. The parameters discussed below identify the bleed rate for each region, for a specific solution. If a bleed region is not named in this file, its bleed rate is set to zero.

There are five possible bleed models, as follows:

```
BLEED ibrg blv1
```

*ibrg*    Bleed region number from *.cgd* file

*blv1*    Normalized bleed flow rate

*blv1* can also be thought of as the mass flow ratio for the bleed region. If  $A_c$  is the reference area prescribed by the user, the actual bleed mass flow is calculated as

$$\dot{m}_b = blv1(\rho_\infty U_\infty A_c)$$

Note:  $A_c$  is an arbitrarily set reference area that you must specify in each zone of the grid file that contains bleed areas.

```
BLEED POROSITY ibrg blv1 blv2 blv3
```

*ibrg*    Bleed region number from *.cgd* file

*blv1*    Back pressure in psia

*blv2*    Porosity

*blv3*    Discharge coefficient

```
BLEED MODEL ibrg mode blv1 blv2 blv3 blv4
```

This keyword specifies use of the empirical bleed model of Mayer and Paynter ([1994](#)).

The input parameter *ibrg* is the bleed region number from the *.cgd* file. The input data for the bleed model is given by the values of *blv1* through *blv4*. Various combinations of values may be specified, depending on the *mode*, as described below.

| <u>mode</u> | <u>blv1</u> | <u>blv2</u> | <u>blv3</u> | <u>blv4</u> |
|-------------|-------------|-------------|-------------|-------------|
| 1           | $p_{plen}$  | Porosity    | $qsmode$    | $N_{bl}$    |
| 2           | $p_{plen}$  | Porosity    | $qsmode$    |             |
| 3           | $p_{plen}$  | Porosity    |             | $M$         |
| 4           | $Q_{sonic}$ | Porosity    | $qsmode$    | $M$         |

In the above table,  $p_{plen}$  is the bleed plenum static pressure,  $N_{bl}$  is the number of grid points in the boundary layer,  $Q_{sonic}$  is the sonic mass flow coefficient (described below), and  $M$  is the local Mach number at the edge of the boundary layer. The parameter  $qsmode$  is an integer from 1 to 3 defining how  $Q_{sonic}$  is to be computed, as follows:

- 1 Set  $Q_{sonic} = 1$
- 2 Compute  $Q_{sonic}$  for 90° holes
- 3 Compute  $Q_{sonic}$  for 20° holes

In the Mayer-Paynter model, the bleed velocity is given by the formula

$$V_{bleed} = Q_{sonic} \Phi \Gamma T \frac{p_T}{p} \frac{1}{\sqrt{T_T}}$$

where  $\Phi$  is the porosity,  $p$  and  $T$  are the local static pressure and temperature, and  $p_T$  and  $T_T$  are the local total pressure and temperature. “Local” means at the edge of the boundary layer at the location of the grid point within the bleed region. The parameter  $\Gamma$  is a function of the specific heat ratio  $\gamma$ .

$$\Gamma = \left(1 + \frac{\gamma - 1}{2}\right)^{-\frac{\gamma+1}{2(\gamma-1)}}$$

Central to the model is  $Q_{sonic}$ , the sonic mass flow coefficient, defined as

$$Q_{sonic} = \frac{\dot{m}_{bleed}}{\dot{m}_{max}} = f\left(\alpha, M, \frac{p_{plen}}{p_T}\right)$$

where  $\dot{m}_{bleed}$  is the actual bleed flow rate and  $\dot{m}_{max}$  is the maximum theoretical bleed flow rate.

$Q_{sonic}$  is a function of the bleed hole angle  $\alpha$ , the local Mach number  $M$ , and the ratio of the plenum pressure  $p_{plen}$  to the local pressure  $p$ . The functional relationship is in the form of tabulated experimental data for circular bleed holes at angles of 20° (McLafferty and Ranard, 1958) and 90° (Syberg and Hickox, 1972).

BLEED FORCING *ibrg blv1 blv2 blv3*

This mode allows an oscillating normal velocity bleed boundary condition to be specified.

- ibrg*    Bleed region number from *.cgd* file
- blv1*    Amplitude of the normal velocity oscillation (ft/sec)
- blv2*    Frequency of the oscillation (Hz)
- blv3*    Phase offset of the oscillation (deg)

BLEED AEDC *ibrg*

- ibrg*    Bleed region number from *.cgd* file

When the AEDC keyword is specified, the bleed region uses the AEDC wind tunnel wall correction correlations. (Note: the entire boundary must be flagged as bleed, not only the porous region.) This condition is hard-wired for the  $j = 1$  boundary and the AEDC wind tunnel and is *not intended for general use*.

*See Also:* [BLOW](#), [MASS FLOW](#)

**BLOW** — Inject vectored flow over a selected region

```
BLOW {ibreg mdot  $T_{inj}$  angle | \
      PLENUM ibreg  $P_t$   $T$  angle}
```

Porous wall cooling over a selected region can be simulated using this keyword. The region must be identified as a bleed region in the grid file. This option is intended for mass inflow only (i.e.,  $P_t > \text{local } P_s$ ). It won't work well for grids that are skewed at the wall, resulting in blowing mass flow errors.

Blowing is currently implemented for  $j$ -boundaries only, and the flow must be in the  $+x$  direction ( $u > 0$ ).

There are two possible blowing modes, as follows:

```
BLOW ibreg mdot  $T_{inj}$  angle
```

*ibreg*    Bleed region number from *.cgd* file  
*mdot*    Injected mass flow in region *ibreg* (lb<sub>m</sub>/sec)  
 $T_{inj}$     Static temperature of injected flow (°R)  
*angle*    Blowing angle relative to  $x$ - $y$  plane (degrees)

```
BLOW PLENUM ibreg  $P_t$   $T$  angle
```

*ibreg*    Bleed region number from *.cgd* file  
 $P_t$     Plenum total pressure (psi)  
 $T$     Plenum static temperature (°R)  
*angle*    Blowing angle relative to  $x$ - $y$  plane (degrees)

If the flowfield static pressure  $P_s$  becomes greater than the plenum total pressure  $P_t$ , the plenum total pressure will be automatically increased to  $1.005P_s$  to maintain a blowing boundary condition. Setting [TEST 52](#) will notify the user when this occurs.

See Also: [BLEED](#), [TEST 52](#)

**BL\_INIT** — Boundary layer initialization
**BL\_INIT** *nb ixs str igrs [delt] nbli nzone*

The **BL\_INIT** keyword signifies the user wishes to initialize a  $j$  or  $k$  boundary with a laminar or turbulent flat plate boundary layer. The user-specified values are defined as:

|              |                                                                                                                 |
|--------------|-----------------------------------------------------------------------------------------------------------------|
| <i>nb</i>    | A number from 1 to 4 indicating which boundary to initialize                                                    |
| 1            | for $j = 1$                                                                                                     |
| 2            | for $j = j_{max}$                                                                                               |
| 3            | for $k = 1$                                                                                                     |
| 4            | for $k = k_{max}$                                                                                               |
| <i>ixs</i>   | Integer indicating method for specifying the axial location to start the boundary layer:                        |
| 1            | to specify the $i$ index                                                                                        |
| 2            | to specify the $x$ station                                                                                      |
| <i>str</i>   | $i$ index or $x$ location to start the boundary layer, depending on the value of <i>ixs</i>                     |
| <i>igrs</i>  | Integer indicating method for specifying the initial height of the boundary layer:                              |
| 1            | to specify the number of grid points from the wall                                                              |
| 2            | to specify the thickness                                                                                        |
| 3            | to compute thickness from initial $x$ station                                                                   |
| <i>delt</i>  | Number of grid points in boundary layer (if $igrs = 1$ ), or boundary layer thickness (inches) (if $igrs = 2$ ) |
| <i>nbli</i>  | Integer indicating type of boundary layer and whether to keep constant thickness or grow with $x$               |
| 1            | Turbulent profile, growing with $x$                                                                             |
| -1           | Turbulent profile, constant thickness                                                                           |
| 2            | Laminar profile, growing with $x$                                                                               |
| -2           | Laminar profile, constant thickness                                                                             |
| <i>nzone</i> | Zone number for initialization                                                                                  |
| 0            | All zones                                                                                                       |
| > 0          | Zone <i>nzone</i> only                                                                                          |

A flat plate temperature distribution will also be constructed consistent with the wall temperature boundary condition of either adiabatic or constant temperature wall.

Note: **BL\_INIT** cannot currently do more than one boundary in each zone without over-writing itself.



**BOUNDARY-DAMP | BDAMP** — Boundary damping (block)

```
{BOUNDARY-DAMP | BDAMP}
  ZONE nzn1:nzn2
  BOUNDARY bnd1:bnd2
  [POINTS n]
  [SECOND c2]
  [FOURTH c4]
  [NOSMOOTHING [NI m1] [NJ m2] [NK m3]]
{END-BOUNDARY-DAMP | ENDBDAMP}
```

This keyword instructs WIND to utilize second- and fourth-order smoothing to damp waves in the vicinity of computational boundaries. It is also possible to turn off smoothing near any boundary for any numerical operator ( $\xi$ ,  $\eta$ , or  $\zeta$ ). This option must be used with the [SMOOTHING](#) keyword and [TEST 49 2](#). The user-specified values are defined as:

- nzn1* Starting zone (0 to begin at zone 1)
- nzn2* Ending zone (0 to end at last zone)
- bnd1* Starting boundary (0 to begin at boundary 1). The values 1–6 correspond to the  $i = 1$ ,  $i = i_{max}$ ,  $j = 1$ ,  $j = j_{max}$ ,  $k = 1$ , and  $k = k_{max}$  boundaries, respectively.
- bnd2* Ending boundary (0 to end at last boundary)
- n* Number of points into the domain over which boundary damping is to be applied
- c2* Second-order smoothing coefficient
- c4* Fourth-order smoothing coefficient

The NOSMOOTHING keyword is used to turn off smoothing (as specified using the [SMOOTHING](#) keyword) for the most recently specified ZONE and BOUNDARY. The keywords NI, NJ, and/or NK specify that smoothing is to be turned off for the  $\xi$ ,  $\eta$ , and/or  $\zeta$  operator, respectively, for boundaries *bnd1* through *bnd2*.

- m1*, *m2*, *m3* Number of points into the domain on boundaries *bnd1* through *bnd2* over which smoothing is to be turned off

Example

```
BOUNDARY-DAMP
  ZONE 2:4 BOUNDARY 1:3 POINTS 15 SECOND .01 FOURTH .03
  ZONE 3:3 BOUNDARY 4:4 POINTS 10 SECOND .02 FOURTH .01
  ZONE 5:5
  BOUNDARY 2:2
  NOSMOOTHING NI 31 NJ 21
END-BOUNDARY-DAMP
```

**BOUNDARY TVD** — Boundary total variation diminishing operator flag

```
BOUNDARY TVD [OFF | FACTOR factor] [zone_selector] \
[BOUNDARY] [ALL | I1 | IMAX | J1 | JMAX | K1 | KMAX | OVERLAP]
```

This keyword controls the TVD flux limiter in the explicit operator at specified coupled boundaries.

**FACTOR** specifies the ‘compression’ parameter for the TVD operator. The TVD operator controls the amount of limiting (larger numbers, less limiting towards first order operator, zero factor is equivalent to first order, one corresponds to no overshoots, two allows some increase in interface value over surrounding values).

The boundary TVD limiter must be more limiting than the interior TVD limiter for it to have any effect. I.e., the value *factor* must be no larger than the value used for the internal scheme. (See the [TVD](#) keyword for the default interior values.)

By default, boundary TVD is on, with a compression parameter equal to that used for the internal scheme.

See Also: [TVD](#)

**CFL#** — CFL/time step specification

```
{CFL# | TIMESTEP} [CFL | SECONDS] {cfl | \
  INCREMENT cfl1 cflmax cflfac inccfl irstart} [zone_selector [ityp]]
```

This keyword allows the user to specify the CFL number or time step for all zones, or on a zone-by-zone basis. All keywords and parameters must be on one line in the data file.

- |         |                                                                                                        |
|---------|--------------------------------------------------------------------------------------------------------|
| CFL     | A CFL number is being specified. This is the default, and should be used for steady flow problems.     |
| SECONDS | The time step is being specified in seconds. This option is generally used for unsteady flow problems. |

If the keyword parameter INCREMENT is not used, the CFL number or time step is given by the input value *cfl*.

If INCREMENT is specified, the CFL number or time step will be gradually increased as the calculation proceeds, based on the following input values:

- |                |                                                                                                                                                                                   |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>cfl1</i>    | CFL number or time step on first iteration                                                                                                                                        |
| <i>cflmax</i>  | Maximum CFL number or time step allowed                                                                                                                                           |
| <i>cflfac</i>  | Factor which multiplies the CFL number or time step every <i>inccfl</i> iterations                                                                                                |
| <i>inccfl</i>  | Iteration increment at which the CFL number or time step is multiplied by <i>cflfac</i> . The multiplication is done whenever $\text{mod}(\text{iteration}, \text{inccfl}) = 1$ . |
| <i>irstart</i> | Starting iteration in the increment calculation                                                                                                                                   |

When a CFL number is being specified, the value of *ityp* may be set to 1 to indicate that a global time step (i.e., constant in space) should be used, equal to the minimum value in the zone. This allows the time step to be determined through a CFL number for unsteady flow problems. Currently, this option is only available for single-zone grids. When a CFL number is specified, and *ityp* = 0 or is omitted, a local time step (i.e., varying in space) will be used.

See Also: [CROSSFLOW](#)

## CGNSBASE — Use CGNS files

|                          |
|--------------------------|
| CGNSBASE <i>basename</i> |
|--------------------------|

This keyword specifies that the grid and flow data are obtained from, and written to, CGNS (CFD General Notation System) files, not common files. The user-specified *basename* is the name of the `CGNSBase_t` node in the grid and flow files.<sup>24</sup>

It should be noted that, strictly speaking, there is no notion of a CGNS *file*, only of a CGNS *database* implemented within one or more ADF (Advanced Data Format) files. We nevertheless use the terminology “CGNS file” in the WIND documentation. In addition, a true CGNS database has a single `CGNSBase_t` node as its root. In the current WIND implementation, however, the grid and flow solution are stored in separate files, with both files having their own `CGNSBase_t` node with the same *basename*.

If the `CGNSBASE` keyword is not used, the grid and flow data are assumed to be in common files.

WIND supports the most common CGNS features, but not all. Some specific limitations on the implementation of CGNS in the WIND code are listed below.

- Many WIND input options may be specified on a zonal basis, using the zone number to identify the zone. In the CGNS standard, zones are identified by a name. When CGNS files are used in WIND, it is assumed that the zone number maps to the zone's position in an alphanumerically sorted list of zone names.

Note that the naming convention `Zone $n$` , where  $n$  is the zone number, is alphanumeric only up to `Zone9`. `Zone10` through `Zone19` would get sorted between `Zone1` and `Zone2`, and so on. However, spaces are allowed in names, so “`Zone  $n$` ”, with two spaces, (e.g., `Zone 1`, `Zone 2`, ..., `Zone 99`, `Zone100`, ...) is alphanumeric up to `Zone999`.

- The CGNS default names “`GridCoordinates`” and “`FlowSolution`” are used for the CGNS `GridCoordinates_t` and `FlowSolution_t` nodes. For each zone, only one grid and one flow solution are stored.
- In a CGNS database, the scaling data must be stored with the data itself, but the units to be used may be stored with the data or higher in the node tree. The WIND implementation assumes that the units are stored with the data.
- If grid velocities are stored in the grid file, it is assumed that the units are ft/sec.
- The CGNS nodes for storing time-dependent data are not yet supported in WIND.
- For overlapping grids, CGNS allows hole points to be identified using multiple `OverSetHoles_t` nodes. WIND currently assumes that all hole points in a zone are identified in a single `OverSetHoles_t` node, using a `PointList`.
- WIND writes convergence and time history information to the list output (*.lis*) and time history (*.cth*) files, respectively, not to a CGNS file.

There are also some WIND features and capabilities that are not supported by the current CGNS standard. These include [chemistry](#) and [MFD](#) flows, [specified boundary layer transition](#), and [bleed boundary conditions](#). Several proposals are pending to extend the CGNS standard, but for now CGNS files created by WIND for these types of flows may not be fully CGNS compliant, and should be considered unique to the WIND implementation.

<sup>24</sup>Detailed information on the CGNS standard may be found at the CGNS web site, at <http://www.cgns.org/>.

**CHEMISTRY** — Chemistry model selection (block)

```

CHEMISTRY
  FILE {filename [LOCAL]}
  {FROZEN | FINITE RATE}
  [FUEL AIR RATIO value]
  [LIU-VINOKUR 2]
  MASS [FRACTIONS] {AIR | sp1 sp2 ... spn}
  OMIT [THIRD-BODY] S1 S2 ... Sm [zone_selector]
  [VISCOSITY {SUTHERLAND | WILKES | KEYE}]
ENDCHEMISTRY

```

This option allows the user to specify the real gas chemistry mode and input data for the desired species and reactions.

Note: The [ARBITRARY INFLOW](#) keyword block, if used, must come after the **CHEMISTRY** keyword block in the input data (*.dat*) file.

The various elements of the **CHEMISTRY** input block are defined as follows:

**CHEMISTRY**

Defines the beginning of the **CHEMISTRY** block

**FILE** {filename [LOCAL]}

Specifies the chemistry data file containing the thermodynamic data, finite rate coefficients, and transport property data for Wilkes' law. (See [Section 7.10](#) for the file format.) If the chemistry data file is located in the directory that WIND files (*.cfl*, *.cgd*, etc.) are running the keyword **LOCAL** must be specified.

{FROZEN | FINITE RATE}

**FROZEN**                Selects frozen chemistry

**FINITE RATE**        Selects finite rate chemistry

**FUEL AIR RATIO** value

value specifies the fuel-air ratio for H<sub>2</sub>-air combustion (two-reaction global model).

**LIU-VINOKUR 2**

Selects the Liu and Vinokur equilibrium air curve fits. Valid to 50000 K, does not output species.

**MASS [FRACTIONS]** {AIR | sp<sub>1</sub> sp<sub>2</sub> ... sp<sub>n</sub>}

Specifies the freestream (reference) mass fractions of the species. The keyword **AIR** indicates use of air mass fractions.

OMIT [THIRD-BODY]  $S_1$   $S_2$  ...  $S_m$  [zone\_selector]

This keyword allows control of the reacting species in a finite-rate chemistry solution on a zonal basis. Reactions leading to the production or destruction of the species given by  $S_1$  through  $S_m$  will be eliminated in the selected zones. In addition, if the **THIRD-BODY** keyword parameter is used, *any* reaction involving the named species will be eliminated. The species will still be convected and diffused, maintaining conservation at zone boundaries.

Multiple **OMIT** keywords are allowed, with conflicting input resolved by using the latter of the conflicting entries.

Eliminating reactions in zones where they aren't relevant improves the overall efficiency of the calculation in two ways. First, the time required to perform the unnecessary computations is eliminated. And second, the stiffness of the system of equations is reduced, leading to faster convergence and greater stability.

**VISCOSITY**

Selects the transport property equations.

**ENDCHEMISTRY**

Defines the end of the **CHEMISTRY** block.

#### General and Standard Chemistry Packages

The chemistry data files supplied with the WIND code are summarized in [Table 1](#).<sup>25</sup>

---

<sup>25</sup>In [Table 1](#), the symbol  $e^-$  is electron density, and  $\text{NO}^+$  is ionized nitrogen oxide. Also,  $T_{max}$  is the maximum temperature at which the curve fits are valid,  $Nc$  is the number of thermodynamic curve fit segments,  $Nr$  is the number of reaction equations, and  $3d$  denotes Variable or Average 3rd-body efficiency.

Table 1: General and Standard Chemistry Packages

| Species                                                                               | File(s)                                                                                      | $T_{max}$         | $Nc$   | $Nr$ | $3d$ |
|---------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|-------------------|--------|------|------|
| O <sub>2</sub> , NO, O, N, N <sub>2</sub>                                             | <i>air-5sp-std-06k.chm</i><br><i>air-5sp-gen-06k.chm</i>                                     | 6000 K            | 1      | 5    | V    |
| O <sub>2</sub> , NO, O, N, N <sub>2</sub>                                             | <i>air-5sp-std-15k.chm</i><br><i>air-5sp-gen-15k.chm</i>                                     | 15000 K           | 3      | 5    | V    |
| O <sub>2</sub> , NO, O, N, N <sub>2</sub>                                             | <i>air-5sp-std-30k.chm</i><br><i>air-5sp-gen-30k.chm</i>                                     | 30000 K           | 5      | 5    | V    |
| O <sub>2</sub> , NO, O, N, NO <sup>+</sup> , e <sup>-</sup> , N <sub>2</sub>          | <i>air-7sp-std-06k.chm</i>                                                                   | 6000 K            | 1      | 6    | V    |
| O <sub>2</sub> , NO, O, N, NO <sup>+</sup> , e <sup>-</sup> , N <sub>2</sub>          | <i>air-7sp-std-15k.chm</i><br><i>air-7sp-gen-15k.chm</i>                                     | 15000 K           | 3      | 6    | V    |
| O <sub>2</sub> , NO, O, N, NO <sup>+</sup> , e <sup>-</sup> , N <sub>2</sub>          | <i>air-7sp-std-30k.chm</i><br><i>air-7sp-gen-30k.chm</i>                                     | 30000 K           | 5      | 6    | V    |
| O <sub>2</sub> , OH, H <sub>2</sub> , H <sub>2</sub> O, N <sub>2</sub>                | <i>h2air-5sp-std-06k.chm</i><br><i>h2air-5sp-std-15k.chm</i>                                 | 6000 K<br>15000 K | 1<br>3 | 2    |      |
| O <sub>2</sub> , H, H <sub>2</sub> , H <sub>2</sub> O, N <sub>2</sub>                 | Not yet available                                                                            |                   |        |      |      |
| O <sub>2</sub> , H, H <sub>2</sub> , H <sub>2</sub> O, OH, O, N <sub>2</sub>          | <i>h2air-7sp-std-15k.chm</i><br><i>h2air-7sp-gen-15k.chm</i><br><i>h2air-7sp-bak-15k.chm</i> | 15000 K           | 3      | 8    |      |
| O <sub>2</sub> , CO <sub>2</sub> , H <sub>2</sub> O, NO <sub>2</sub> , N <sub>2</sub> | Not yet available                                                                            |                   |        |      |      |

/ — Comment lines

In the extended portion of the file, a “comment line” may be placed anywhere. This line is denoted by a / in column 1. The remainder of the line is disregarded by the input routines, as if the comment line did not exist. The purpose of this option is to allow the user to place explanatory notes within the flow condition data file. A blank line is also ignored.



**COMPRESSOR FACE** — Outflow boundaries, compressor face

```
COMPRESSOR FACE {CHUNG mach zone | BOEING mode val1 [val2] zone | \
                  PAYNTER mach val1 val2 zone | SAJBEN mach angh ang zone}
```

This keyword allows an outflow boundary to be modeled as a compressor face. There are several models available, as described below. All are based on the observation that turbine engine conditions set the corrected mass flow, and that this corresponds directly to the average Mach number at the compressor face. These boundary conditions have been implemented mainly for the analysis of unsteady flow; however, they have also been shown to be robust for the establishment of steady-state, supercritical inlet flows.

```
COMPRESSOR FACE CHUNG mach zone
```

The CHUNG keyword option uses the Chung-Cole model (Chung and Cole, 1996). Based on the specified mass-averaged Mach number *mach*, and using the computed local Mach number and total pressure, a new local static pressure is computed.

Note: This boundary condition is identical in its effect on the flow to the DOWNSTREAM MACH keyword.

```
COMPRESSOR FACE BOEING mode val1 [val2] zone
```

The BOEING keyword option uses a model developed by Mayer and Paynter (1994). The parameters to be specified as *val1* and *val2* depend on *mode*, as shown in the following table.

| <u>mode</u> | <u>val1</u>                                                                | <u>val2</u>                   |
|-------------|----------------------------------------------------------------------------|-------------------------------|
| 1           | Average Mach number                                                        | —                             |
| 2           | Corrected mass flux per unit area,<br>lb <sub>m</sub> /sec/ft <sup>2</sup> | —                             |
| 3           | Nominal Mach number                                                        | Nominal total temperature, °R |

Modes 1 and 2 are equivalent, and are useful in obtaining initial flow fields, and for simulating instantaneous changes at the compressor face via a restart. Mode 3 is used to set a constant volumetric flow condition. Typically, an initial condition is obtained using mode 1 and becomes the nominal condition. Then *val1* is set to the Mach number used in mode 1 and *val2* is set to the freestream total temperature, i.e., it is assumed that the total temperature at the compressor face is equal to the freestream value. If the average total temperature at the compressor face does not change, then modes 1 and 3 will yield identical results. However, if the average total temperature at the compressor face changes due to a freestream disturbance, for example, then the corrected flow rate will be adjusted to maintain a constant volumetric flow rate.

```
COMPRESSOR FACE PAYNTER mach val1 val2 zone
```

The Paynter compressor face model (Paynter, 1998) computes unsteady acoustic reflections from a compressor face subjected to acoustic and convective disturbances. This model is intended to be used for time-accurate simulations of unsteady inlet flow. The CFL number should be less than one. The flow is assumed to be subsonic and axial as it enters the compressor face.

The model assumes the compressor face is positioned within an annular duct with a hub and case, and consists of a single row of axial flow compressor blades. The conditions at the compressor face

are determined by the convective velocity response coefficient  $\alpha$ , and the acoustic response coefficient  $\beta$ . The definitions of  $\alpha$  and  $\beta$  depend on the passage Mach number  $M_{\text{passage}}$ , which is given by

$$M_{\text{passage}} = M_{cf} / \cos \Gamma$$

where  $M_{cf}$  is the compressor face Mach number, and  $\Gamma$  is the local stagger angle.

For subsonic flow ( $M_{\text{passage}} < 1$ ),

$$\alpha = \frac{\gamma M_{cf}^2}{1 - M_{cf}^2} \tan \left( \frac{\Gamma}{2} \right) \tan \Gamma$$

$$\beta = \tan^2 \left( \frac{\Gamma}{2} \right) \left( \frac{1 + M_{cf}^2}{1 - M_{cf}^2} \right)$$

where  $\gamma$  is the specific heat ratio. For supersonic flow ( $M_{\text{passage}} > 1$ ),

$$\alpha = \gamma M_{cf}$$

$$\beta = 1$$

Details on the implementation of this boundary condition are given by [Slater and Paynter \(2000\)](#).

In the WIND code,  $\alpha$  and  $\beta$  may be treated as constant and specified directly, or computed using the above equations. The local stagger angle  $\Gamma$  is determined by assuming a linear variation between specified values of the stagger angles at the hub and case. The solidity of the blade row is assumed to be greater than one.

The absolute value of the keyword parameter *mach* sets the nominal or average compressor face Mach number  $M_{cf}$ . The meanings of *val1* and *val2* depend on the sign of *mach*:

| <i>mach</i> | <i>val1</i>                                                    | <i>val2</i>                                                     |
|-------------|----------------------------------------------------------------|-----------------------------------------------------------------|
| < 0         | Stagger angle (degrees) of the compressor fan blade at the hub | Stagger angle (degrees) of the compressor fan blade at the case |
| > 0         | Constant convective velocity response coefficient, $\alpha$    | Constant convective velocity response coefficient, $\beta$      |

COMPRESSOR FACE SAJBEN *mach angh angc zone*

The Sajben compressor face model is basically the same as the Paynter model, except that Sajben's expression for the acoustic response coefficient is used for subsonic flow ([Sajben, 1999](#)).

Sajben's expression for the acoustic response coefficient may be written as

$$\beta = \frac{M_{cf} \tan^2 \Gamma}{2(1 + M_{cf}^2) + M_{cf}^2 \tan^2 \Gamma}$$

where  $M_{cf}$  is the compressor face Mach number, and  $\Gamma$  is the local stagger angle.

With the COMPRESSOR FACE SAJBEN keyword, the keyword parameter *mach* is the nominal or average compressor face Mach number  $M_{cf}$ , and *angh* and *angc* are the stagger angles in degrees of the compressor fan blade at the hub and case.

### Examples

```
COMPRESSOR FACE CHUNG 0.3 1  
COMPRESSOR FACE BOEING 1 0.3 1  
COMPRESSOR FACE BOEING 2 0.5516 1  
COMPRESSOR FACE BOEING 3 0.3 520.0 1
```

*See Also:* [DOWNSTREAM MACH](#), [DOWNSTREAM PRESSURE](#), [MASS FLOW](#)

### CONVERGE — Controls convergence

|                                     |
|-------------------------------------|
| CONVERGE {LEVEL   ORDER} <i>val</i> |
|-------------------------------------|

This keyword allows the user to specify the convergence criteria used for a run. The two possible options are defined below.

LEVEL    Check for maximum residual less than *val*.

ORDER    Check for reducing maximum residual by *val* orders of magnitude.

The default is a four-order of magnitude decrease in the maximum residual.

If Newton iteration is being used, this keyword specifies the convergence criteria within a Newton time step. The overall global convergence criteria is specified using the [NEWTON](#) keyword.

*See Also:* [TEST 128](#), [NEWTON](#)

**COUPLING** — Zone coupling mode specification

|                                                                    |
|--------------------------------------------------------------------|
| COUPLING [MODE] {CHARACTERISTIC   <u>ROE</u> [ <u>HIGH</u>   LOW]} |
|--------------------------------------------------------------------|

This keyword controls the zone coupling algorithm used for multi-zone solutions. The two possible options are defined below.

**CHARACTERISTIC** This is NASTD's original zone coupling algorithm that uses one-dimensional characteristic flow theory to set boundary flowfield variables based on local flow direction and strength. These boundary variables are then transferred between zones using interpolation factors stored in the grid file (the *.cgd* file).

**ROE** This newer zone coupling algorithm is more consistent with Roe's flux-difference splitting scheme, which is WIND's default explicit operator. Instead of transferring flowfield variables between zones, this algorithm transfers flux cell "interface states," which are critical quantities in Roe's scheme. In this coupling mode, the zone boundary may be thought of as a "cell interface" in Roe's scheme.

The **HIGH** option, which is the default, turns on higher-order coupling at zone boundaries. With higher-order coupling the solution derivatives are also passed between coupled zones. This results in a slight increase in memory requirements. The derivative information is used to increase the accuracy of the coupling scheme and to improve robustness at coupled boundaries through the application of a TVD operator.

The default **HIGH** option is applicable to the following explicit operators, selectable using the **RHS** keyword:

- Second-order Roe or Van Leer fully upwind  
(**RHS ROE/VAN LEER SECOND**)
- Third-order Roe or Van Leer fully upwind  
(**RHS ROE/VAN LEER THIRD**)
- Second-order Roe or Van Leer central  
(**RHS ROE/VAN LEER CENTRAL**)
- Second-order Roe or Van Leer mixed upwind and central  
(**RHS ROE/VAN LEER MIXED**)
- First-order Roe upwind modified for stretched grids  
(**RHS PHYSICAL FIRST**)
- Second-order Roe upwind modified for stretched grids  
(**RHS PHYSICAL SECOND**)

If some other explicit operator is being used, you must specify **COUPLING MODE ROE LOW**, or **COUPLING MODE CHARACTERISTIC**.

See Also: **RHS**

**CYCLES** — Number of solution cycles

**CYCLES** *ncyc*

This keyword is used to set the number of cycles *ncyc*, and is required in the input data file. By default each cycle has five flowfield iterations in all zones, but this may be changed using the [ITERATIONS](#) keyword.

Cycle time information will be written to the *.lis* file every *freq* cycles, starting with the first cycle. The default is every cycle. Residuals will also be written to the *.lis* file every *freq* cycles, starting with the first cycle, but only for iterations consistent with the **PRINT FREQUENCY** specified with the [ITERATIONS](#) keyword.

Note that this does not affect integrated properties specified using the [LOADS](#) keyword block. The print frequency for integrated properties is controlled by the **PRINT** keyword in the **LOADS** keyword block.

### Example

If the user specifies

```
CYCLES 1000 PRINT FREQUENCY 10
ITERATIONS PER CYCLE 10 PRINT FREQUENCY 10
```

cycle time information will be printed for cycles 1, 11, 21, etc. Residuals will be printed for iterations 10 (in cycle 1), 110 (cycle 11), 210 (cycle 21), etc.

See Also: [ITERATIONS](#), [LOADS](#)

**CROSSFLOW** — Crossflow CFL factor

|                                                              |
|--------------------------------------------------------------|
| CROSSFLOW [CFL] [FACTOR] <i>val</i> [ <i>zone_selector</i> ] |
|--------------------------------------------------------------|

The **CROSSFLOW** keyword sets the cross-flow CFL factor to *val* in the specified zones.

For most WIND calculations, the time step is based on a constant CFL number throughout the flow field. Thus, for non-uniform grids, the solution advances at different rates in different parts of the grid. Because the CFL number is directly proportional to the physical time step and inversely proportional to the local grid spacing, the physical time step is small where the grid spacing is small, in order to maintain a constant CFL number. For example, in boundary layers and shear layers, where the grid is often closely packed for better resolution, the solution advances more slowly than in other parts of the flowfield. In fact, most of the iterations in a viscous solution are spent converging viscous regions.

To speed up convergence in these regions, each coordinate direction's eigenvalues are multiplied by a factor before determining the time step associated with the local spacing and flow conditions. This factor varies from 1.0 when the flow is along the coordinate direction, to the specified value when the flow is perpendicular to the coordinate direction. The far-field CFL number is generally the value specified in the input data file with the **CFL#** keyword, but near the wall the code effectively increases the time step in the boundary layer by the specified factor. This should increase convergence of the boundary layer.

The default cross-flow CFL factor is 2.0 for three-dimensional flow, and 1.0 (i.e., no time step increase in the boundary layer) for two-dimensional flow and axisymmetric flow.

*See Also:* **CFL#**

**DOWNSTREAM MACH** — Outflow boundaries, Mach number

|                                           |
|-------------------------------------------|
| DOWNSTREAM MACH [NUMBER] <i>mach zone</i> |
|-------------------------------------------|

This keyword allows the specification of a mass-averaged subsonic Mach number *mach* at outflow boundaries. Based on the specified mass-averaged Mach number, and using the computed local Mach number and total pressure, a new local static pressure is computed. This boundary condition simulates the uniform Mach number characteristics observed at the compressor faces of turbine engines, and is identical in its effect on the flow to the **COMPRESSOR FACE CHUNG** keyword.

*See Also:* [COMPRESSOR FACE](#), [DOWNSTREAM PRESSURE](#), [MASS FLOW](#)



**DOWNSTREAM PRESSURE** — Outflow boundaries, pressure

```
DOWNSTREAM PRESSURE {value [EXTRAPOLATE [SUPERSONIC]] zone | \
FREESTREAM [EXTRAPOLATE [SUPERSONIC]] zone | \
EXTRAPOLATE [ALWAYS] zone | \
value VARIABLE i j k zone | \
value UNSTEADY {SINUSOIDAL | USERSPEC}  $\Delta p$  freq phase ZONE zone}
```

This keyword controls the specification of a pressure boundary condition at outflow boundaries. There are five possible modes, defined as follows:

```
DOWNSTREAM PRESSURE value [EXTRAPOLATE [SUPERSONIC]] zone
```

This mode specifies the pressure *value* to be applied at the outflow boundary in the specified zone. The EXTRAPOLATE [SUPERSONIC] option requests that WIND extrapolate the pressure from the point adjacent to the boundary for supersonic flow points.

```
DOWNSTREAM PRESSURE FREESTREAM [EXTRAPOLATE [SUPERSONIC]] zone
```

This mode is the same as the previous one, except that the pressure value is taken from the [FREESTREAM](#) keyword.

```
DOWNSTREAM PRESSURE EXTRAPOLATE [ALWAYS] zone
```

This mode extrapolates the pressure for all points on the boundary, regardless of whether or not they are supersonic.

```
DOWNSTREAM PRESSURE value VARIABLE i j k zone
```

This mode imposes the pressure *value* only at the specified  $(i,j,k)$  boundary point in the specified zone. The indices  $(i,j,k)$  should correspond to a point on the outflow boundary, although at present no check is made to verify that they do. The pressure at the remaining boundary points will spatially vary according to the spatial variation of pressure at the solution points adjacent to the boundary.

```
DOWNSTREAM PRESSURE value UNSTEADY \
{SINUSOIDAL | USERSPEC}  $\Delta p$  freq phase ZONE zone}
```

This mode allows an unsteady pressure to be applied at the outflow boundary in the specified zone. A sinusoidal or user-defined oscillation may be specified. The parameters are defined as:

*value*     Baseline pressure,  $p_s$  (psi)  
 $\Delta p$      Amplitude of oscillation (psi)  
*freq*     Frequency of oscillation,  $\omega$  (Hz)  
*phase*     Phase angle of oscillation,  $\phi$  (deg)

If **SINUSOIDAL** is used, specifying a sinusoidal pressure oscillation, the pressure is computed from

$$p = p_s + \Delta p \sin[2\pi(\omega t + \phi)]$$

where  $p$  is the pressure and  $t$  is the time.

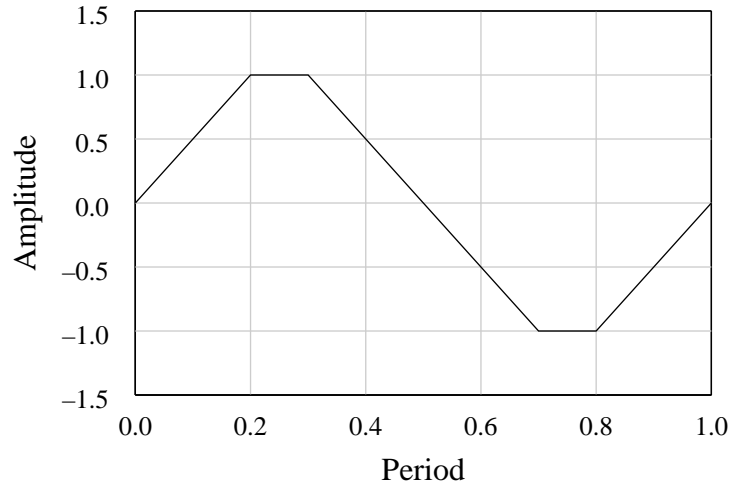
If **USERSPEC** is used, specifying a user-defined pressure oscillation, a table must be input immediately following the **DOWNSTREAM PRESSURE** keyword, with the following format:

```

PROFILE npts
per(1)    amp(1)
per(2)    amp(2)
...
per(npts) amp(npts)

```

where  $npts$  is the number of points in the profile (max 65),  $per$  is the normalized period (i.e., from 0.0 to 1.0), and  $amp$  is the normalized amplitude (i.e., from -1.0 to 1.0). As an example, the following normalized pressure oscillation:



**Figure 7:** Example user-specified unsteady pressure oscillation

could be specified using the following profile input:

```

PROFILE 6
0.0  0.0
0.2  1.0
0.3  1.0
0.7 -1.0
0.8 -1.0
1.0  0.0

```

See Also: [COMPRESSOR FACE](#), [DOWNSTREAM MACH](#), [MASS FLOW](#), [FREESTREAM](#)

**DQ** —  $\Delta Q$  limiter

**DQ** [**LIMITER**] [OFF | ON [**DRMAX** *drmax*] [**DTMAX** *dtmax*]]

This keyword may be used to limit the change in density and/or temperature over a single iteration. This may be helpful, especially in the initial stages of a calculation, in maintaining stability, and allow the use of larger CFL numbers than might otherwise be possible.

**DRMAX** *drmax*    *drmax* specifies the fractional change allowed in the density over an iteration. I.e., specifying “**DQ LIMITER ON DRMAX 0.2**” means the density will not be permitted to change by more than  $\pm 20\%$  from one iteration to the next.

**DTMAX** *dtmax*    *dtmax* specifies the fractional change allowed in the temperature over an iteration.

The default is to not limit the change in density or temperature. When limiting is turned on, the defaults for *drmax* and *dtmax* are both 0.5. Thus, if just “**DQ LIMITER ON**” is specified, both density and temperature will be allowed to change by no more than  $\pm 50\%$  over an iteration.

The use of the **DQ LIMITER** keyword becomes somewhat problematic when multiple sub-iterations are used for the Navier-Stokes equations (using the **NAVIER-STOKES ITERATIONS** keyword) and/or the turbulence model equations (using the **ITERATIONS** option with the **TURBULENCE** keyword). By default, the number of sub-iterations for both the Navier-Stokes and turbulence model equations is one. In this case, each iteration for the Navier-Stokes equations is followed by an iteration for the turbulence model equations, and if the limiter is applied to the Navier-Stokes solution, it is also applied to the turbulence model solution.

If the number of Navier-Stokes sub-iterations is one, and if multiple sub-iterations are used for the turbulence model equations, then if the limiter is applied to the Navier-Stokes solution, it is applied to each turbulence model sub-iteration.

Also, if multiple Navier-Stokes sub-iterations are used, it’s possible that the computed flow field is such that the limiter is applied during some sub-iterations, but not others. In this case, if the limiter is applied during the final Navier-Stokes sub-iteration, it is again applied to each turbulence model sub-iteration. If the limiter is not applied during the final Navier-Stokes sub-iteration, it will not be applied at all during the turbulence model iteration.

For these reasons, it is recommended that the number of sub-iterations for the Navier-Stokes and turbulence model equations be left at their default values of one when the **DQ LIMITER** keyword is used.

**END** — Termination

[END]

This keyword, which must begin in column 1, may be used to end the input data file.

**FIXER** — Instability smoothing

|                                                        |
|--------------------------------------------------------|
| <b>FIXER</b> [ <b>PRINT</b> ] [ <i>zone_selector</i> ] |
|--------------------------------------------------------|

*The **FIXER** keyword is broken in WIND 4, and should not be used. In a given *i*-plane, when any non-active point (i.e., wall, coupled boundary, hole, or fringe point) is found, the search for negative values moves to the next *i*-plane, without examining any remaining points in the original *i*-plane.*

WIND’s **FIXER** capability searches for computational points at which there are negative values of pressure or temperature (due to numerical instabilities) and replaces the faulty values with averages from the nearest “clean” points. This smoothing may help your solution through difficult transients, but *it may also simply smear out instabilities over a larger portion of the flowfield*. You should maintain a careful watch over your solution when using **FIXER**.

The **PRINT** option requests a more descriptive printout of **FIXER** activity.

**FREESTREAM** — Freestream conditions

|                                                                  |
|------------------------------------------------------------------|
| <b>FREESTREAM</b> [STATIC   TOTAL] [CONDITIONS] <i>M P T α β</i> |
|------------------------------------------------------------------|

This keyword allows input of freestream flow conditions. The user-specified parameters are defined as:

*M* Mach number

*P* Static or total pressure (psi), depending on the **STATIC | TOTAL** keyword

*T* Static or total temperature (°R), depending on the **STATIC | TOTAL** keyword

*α, β* Angles of attack and yaw (degrees)

Note that, if *U* is the freestream velocity magnitude, *α* and *β* are related to the Cartesian velocity components by:

$$\begin{aligned}w &= U \sin \beta \\u &= U \cos \beta \cos \alpha \\v &= U \cos \beta \sin \alpha\end{aligned}$$

Note: The local flow angles (*localalpha* and *locbeta* in the **CFPOST** post-processing package) are defined relative to the *x, y, z* coordinates and are given by:

$$\begin{aligned}localalpha &= \tan^{-1}(v/u) \\locbeta &= \tan^{-1}(w/u)\end{aligned}$$

Thus if *β* is not equal to 0, *locbeta* is not equal to *β*.

Note: WIND assumes *α* is rotation about the *z* axis. That is, *y* is up.

**FRINGE** — Solution mode at fringe points

**FRINGE** [MODE] **FROZEN** [*zone\_selector*]

For overlapping zonal boundaries, this keyword may be used to freeze the solution at fringe points during a cycle. Solution values at these points are then updated at the end of the cycle.

**FRINGE FROZEN** *must* be used for double-fringe boundaries. Running a double-fringe grid without this keyword may lead to erroneous results.

**FRINGE FROZEN** may also be specified for single-fringe boundaries, but is not required. The default for single-fringe boundaries is to update the solution at fringe points every iteration.

Using double fringes increases the solution accuracy at overlapping boundaries. Note, though, that by freezing the solution over a cycle, the solution at fringe points will lag in time behind the solution at interior points. This could potentially lead to instabilities during transients, or a reduction in the convergence rate if a large number of iterations per cycle are specified. However, testing on a limited number of cases has not revealed any problems when using up to five iterations per cycle.

Double-fringe boundaries may be created with **GMAN**, by first selecting double-fringe mode using (in GMAN's graphical mode) **MODIFY BNDY > GENERATE FRINGE > SET FRINGE-MOD > DOUBLE FRINGE**, then generating the fringe.

**GAS** — Gas property specification

|                                    |
|------------------------------------|
| <b>GAS</b> <i>gam prl prt gasc</i> |
|------------------------------------|

This option allows the user to input different values of  $\gamma$ , laminar and turbulent Prandtl numbers, and gas constant. The parameters are defined as:

*gam*     Ratio of specific heats  $\gamma = c_p/c_v$  (default is 1.4)

*prl*     Laminar Prandtl number (default is 0.72)

*prt*     Turbulent Prandtl number (default is 0.90)

*gasc*     Gas constant  $R$  (default is 1716 ft<sup>2</sup>/sec<sup>2</sup>-°R)

Note: The [ARBITRARY INFLOW](#) keyword block, if used, must come after the **GAS** keyword in the input data (*.dat*) file.



**GRAVITY** — Add gravity body forces

|                                                    |
|----------------------------------------------------|
| <b>GRAVITY</b> $x_c$ $y_c$ $z_c$ $g_x$ $g_y$ $g_z$ |
|----------------------------------------------------|

The **GRAVITY** keyword allows the user to include the effects of gravity on the fluid.

$x_c, y_c, z_c$     Coordinates of a point on the “ground” plane (normal to the gravity vector) used to define the location of zero potential energy, in grid units

$g_x, g_y, g_z$     Components of the gravity vector, in ft/sec<sup>2</sup>. For example, if the  $y$  direction is normal to the earth’s surface, these values might be 0.0, −32.2, 0.0.

### GRID LIMITER — Grid limiting capability

`GRID LIMITER {ANGLE angle | OFF} [zone_selector]`

WIND's grid limiting capability switches the explicit operator to a first-order scheme in the presence of grid turning greater than a specified amount. This capability may be needed to reduce numerical instabilities near wing tips, leading edges, and other geometric features which may require extreme amounts of grid turning.

The parameter *angle* is the number of degrees off of a straight line allowed before limiting the scheme to first order.

*See Also:* [TEST 110](#), [TVD](#)

**HISTORY** — Time history flowfield variable tracking

```

HISTORY VARIABLE [MACH] [PRESSURE] [TEMPERATURE] [DELTAP] [RADIUS] \
[U_VELOCITY] [V_VELOCITY] [W_VELOCITY] [FREQUENCY f]
HISTORY nzn ibeg iend jbeg jend kbeg kend
HISTORY nzn ibeg iend jbeg jend kbeg kend
...

```

For unsteady flow problems, where the time step is being specified in seconds and is the same throughout the flow field, the **HISTORY** keyword may be used to track flowfield data through the course of a WIND solution. WIND will track the specified variable(s) (at least one must be specified) at the specified locations, with sampling occurring every  $f$  iterations.

The first **HISTORY** line specifies the variable(s) to be tracked and the frequency. The following variables may be tracked:

|                    |                                                                                                                                                                                                                                           |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>MACH</b>        | Mach number                                                                                                                                                                                                                               |
| <b>PRESSURE</b>    | Static pressure                                                                                                                                                                                                                           |
| <b>TEMPERATURE</b> | Static temperature                                                                                                                                                                                                                        |
| <b>DELTAP</b>      | Max difference in pressure from the value at a specified reference point, where the search is made along a line in the $\pm i$ , $\pm j$ , or $\pm k$ direction. (See below for how to specify the reference point and search direction.) |
| <b>RADIUS</b>      | Distance from the specified reference point to the location of <b>DELTAP</b> . The <b>DELTAP</b> and <b>RADIUS</b> parameters are intended for use in tracking the position of a shock wave.                                              |
| <b>U_VELOCITY</b>  | Velocity in the $x$ direction                                                                                                                                                                                                             |
| <b>V_VELOCITY</b>  | Velocity in the $y$ direction                                                                                                                                                                                                             |
| <b>W_VELOCITY</b>  | Velocity in the $z$ direction                                                                                                                                                                                                             |

The additional **HISTORY** lines specify the location. For all variables except **DELTAP** and **RADIUS**, values will be tracked in zone  $nzn$ , in the range  $(ibeg, jbeg, kbeg)$  to  $(iend, jend, kend)$ . For **DELTAP** and **RADIUS**, the reference point is given by  $(ibeg, jbeg, kbeg)$ , and the search direction is defined by setting one of  $(iend, jend, kend)$  to  $\pm 1$ , and the other two values to 0.

Note that only one line may be specified containing history variables, while multiple lines containing ranges may be specified.

The results will normally be stored in a common time history (*.cth*) file, containing a lookup table corresponding to the range of computational indices tracked during the run, time stamp data, and flowfield data for each time stamp. Upon completion of a time history run, the auxiliary program *timplt* may be used to analyze the data and/or to create a GENPLOT file that may then be plotted using CFPOST.

The **HISTORY** keyword is not allowed in parallel mode unless **TEST 125 1** is also specified. With **TEST 125** specified, the time history output is written into the *.lis* file instead of a *.cth* file. This can significantly increase the size of the *.lis* file, but allows time history to be used in parallel mode.

*resplt* can then be used to create a GENPLOT file containing the time history of the specified variables.

The *.cth* file contains reference and scaling data. Thus, when *timplt* is used to create a GENPLOT file, the time history variables will be dimensional. However, when TEST 125 is used to write the time history variables into the *.lis* file, the variables are nondimensional. Thus, when *resplt* is used, the variables in the resulting GENPLOT file will be nondimensional. This is summarized in the following table, showing the values stored in the GENPLOT file using *timplt* and *resplt*. In the table,  $p_\infty$ ,  $T_\infty$ , and  $a_\infty$  are the static pressure, static temperature, and speed of sound at freestream conditions, and  $L_{ref}$  is the reference length used in the grid (*.cgd*) file.

| <u>HISTORY Variable</u> | <u><i>timplt</i></u>                                  | <u><i>resplt</i></u>         |
|-------------------------|-------------------------------------------------------|------------------------------|
| MACH                    | $M$                                                   | $M$                          |
| PRESSURE                | $p$ (lb <sub>f</sub> /in <sup>2</sup> )               | $p/(\gamma p_\infty)$        |
| TEMPERATURE             | $T$ (°R)                                              | $T/T_\infty$                 |
| DELTAP                  | $\Delta p/\gamma$ (lb <sub>f</sub> /in <sup>2</sup> ) | $\Delta p/(\gamma p_\infty)$ |
| RADIUS                  | Distance (in)                                         | Distance/ $L_{ref}$          |
| U_VELOCITY              | $u$ (ft/sec)                                          | $u/a_\infty$                 |
| V_VELOCITY              | $v$ (ft/sec)                                          | $v/a_\infty$                 |
| W_VELOCITY              | $w$ (ft/sec)                                          | $w/a_\infty$                 |

#### Note

An improved time history tracking capability was added to WIND alpha 5.52. This change also required some changes to the shell scripts used to run the WIND code. With the new scripts, the *DATE* suffix that was previously added to the *.cth* file name has been eliminated.

The same scripts are used with all versions of WIND. Thus, if the new scripts are being used with versions of WIND prior to alpha 5.52, when running a restart case you must manually rename any pre-existing *.cth* file for that case. If you don't, the restart case will abort due to a "failure in common I/O library call" when attempting to create the time history file. (The existing *.cth* file will be unchanged.)

The new scripts were uploaded to the IVMS web site on May 30, 2001. Therefore, if you've downloaded and installed the WIND application distribution after that date, you're using the new scripts.

#### Example 1

The following use of the HISTORY keyword will create a time history (*.cth*) file containing the static pressure at the points (17,1,1), (17,11,5), and (9,11,7) in zones 1, 2, and 3, respectively. Values will be written to the file every 10 iterations.

```
History variable pressure frequency 10
History 1 17 17 1 1 1 1
History 2 17 17 11 11 5 5
History 3 9 9 11 11 7 7
```

#### Example 2

In the following example, the Mach number and pressure will be recorded in zone 5, for the grid points  $(i, j, k) = (7-16, 22-24, 19)$ . The values will be written into the *.lis* file every 5 iterations.

```
Test 125 1
History variable Mach pressure frequency 5
```

```
History 5 7 16 22 24 19 19
```

Example 3

In this example, the position of a shock wave will be tracked in the  $+i$  direction in zone 4. The point (1,52,20) will be used as the reference point, and values will be recorded every 20 iterations.

```
History variable radius frequency 20
History 4 1 1 52 0 20 0
```

See Also: [TEST 125](#)

**IMPLICIT** — Implicit operator control

```
IMPLICIT {JACOBI jiter jcnv [RELAX jrel] [zone_selector] | \
          GAUSS giter gcnv [RELAX grel] [zone_selector] | \
          MACCORMACK [MAFK k] [zone_selector] | \
          word1 [word2 word3 [nzn [tre]]]}
```

This keyword allows user control of the implicit operator within each zone. There are four possible modes, defined as follows:

```
IMPLICIT JACOBI jiter jcnv [RELAX jrel] [zone_selector]
```

This form of the **IMPLICIT** keyword turns on the point Jacobi implicit operator in the selected zone. The user-specified parameters are defined as follows:

- jiter*    The number of subiterations allowed each time step. A typical value is 30.
- jcnv*    The level of convergence to assume the subiterations are converged. A typical value is 0.0001.
- jrel*    The relaxation factor. The default value is 1.0 (i.e., no relaxation).

```
IMPLICIT GAUSS giter gcnv [RELAX grel] [zone_selector]
```

This form of the **IMPLICIT** keyword turns on the Gauss-Seidel implicit operator in the selected zone. The user-specified parameters are defined as follows:

- giter*    The number of subiterations allowed each time step. A typical value is 10.
- gcnv*    The level of convergence to assume the subiterations are converged. A typical value is 0.0001.
- grel*    The relaxation factor. The default value is 1.0 (i.e., no relaxation).

```
IMPLICIT MACCORMACK [MAFK k] [zone_selector]
```

This form of the **IMPLICIT** keyword turns on MacCormack's first-order modified approximate factorization (MAFk) procedure in the selected zone. The procedure attempts to remove the decomposition error by feeding the error term back into the matrix equations on the right-hand side.

The solution process requires subiterations, with the number of subiterations specified with the **MAFK** option by the parameter *k*. The default for *k* is 2.

For stability, the **IMPLICIT BOUNDARY** keyword should be used to specify that implicit boundary conditions are to be used with the MAFk procedure. In addition, when large CFL numbers are to be used, it is recommended that the CFL number be increased gradually (over 200 iterations or so) to the desired value using the **INCREMENT** parameter in the **CFL#** keyword.

The MAFk procedure has been demonstrated in WIND to be stable in two dimensions with very high CFL numbers (greater than 1000). In three dimensions, however, only limited testing has been done, and its efficiency has not yet been determined.

**IMPLICIT** *word1* [*word2 word3* [*nzn* [*tre*]]]

This form of the the **IMPLICIT** keyword allows the user to turn off the implicit operator completely, resulting in an explicit calculation, or to use a scalar or full block implicit operator. With these options, a different implicit operator may be specified for each computational direction. This form is also used when the parabolized Navier-Stokes equations are being solved, to specify that the PNS implicit operator should be used in the  $\xi$  direction.

The user-specified parameters are defined as follows:

|              |                                                                                                                                                                                                                                                                              |   |                                     |              |          |     |                               |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|-------------------------------------|--------------|----------|-----|-------------------------------|
| <i>word1</i> | A keyword controlling the $\xi$ -direction implicit operator                                                                                                                                                                                                                 |   |                                     |              |          |     |                               |
| <i>word2</i> | A keyword controlling the $\eta$ -direction implicit operator                                                                                                                                                                                                                |   |                                     |              |          |     |                               |
| <i>word3</i> | A keyword controlling the $\zeta$ -direction implicit operator                                                                                                                                                                                                               |   |                                     |              |          |     |                               |
| <i>nzn</i>   | Specifies the zone. If <i>nzn</i> = 0 or is omitted, the specified operator(s) will be used in all zones not specified with other <b>IMPLICIT</b> keywords in the input file.                                                                                                |   |                                     |              |          |     |                               |
| <i>tre</i>   | Specifies the trapezoidal time differencing factor. The possible values are: <table> <tr> <td>0</td><td>Explicit (but expensive, don't use)</td></tr> <tr> <td>1 or omitted</td><td>Implicit</td></tr> <tr> <td>0.5</td><td>Trapezoidal time differencing</td></tr> </table> | 0 | Explicit (but expensive, don't use) | 1 or omitted | Implicit | 0.5 | Trapezoidal time differencing |
| 0            | Explicit (but expensive, don't use)                                                                                                                                                                                                                                          |   |                                     |              |          |     |                               |
| 1 or omitted | Implicit                                                                                                                                                                                                                                                                     |   |                                     |              |          |     |                               |
| 0.5          | Trapezoidal time differencing                                                                                                                                                                                                                                                |   |                                     |              |          |     |                               |

The keywords that may be used for *word1*, *word2*, and *word3* are defined in the following table.

| <u>Keyword</u> | <u>Difference Scheme</u>                                     |
|----------------|--------------------------------------------------------------|
| NONE           | Explicit                                                     |
| SCALAR         | Scalar implicit operator (default for Euler solutions)       |
| FULL           | Full block implicit operator (default for viscous solutions) |
| PNSX           | PNS implicit operator (marching in $\xi$ )                   |

Note that if only one of the above keywords is specified, WIND assumes that the specified operator is to be used in all directions in all zones.

See Also: [IMPLICIT BOUNDARY](#)

**IMPLICIT BOUNDARY** — Implicit boundary conditions

`IMPLICIT BOUNDARY {ON | OFF} [zone_selector]`

This keyword controls the use of WIND's numerical boundary conditions within the implicit operator. The default mode is `OFF`, implying explicit boundary conditions. When `ON` is specified, implicit boundary conditions will be used at “wall”-type boundaries.

*See Also:* [IMPLICIT](#)



**INCLUDE** — Include a file in the standard input

**INCLUDE** *filename*

This keyword allows the contents of an external file to be incorporated into WIND's standard input, just as if they were part of the input data (*.dat*) file. The *filename* must be either the full path name for the file, or a relative path name within the directory in which the WIND code is being run.

“Include” files may be nested.

**ITERATIONS** — Set number of iterations per cycle

```
{ITERATIONS [PER] [CYCLE] | ITER_CYCLE} n [PRINT FREQUENCY freq] \
[zone_selector]
```

This keyword allows the user to control the number of iterations performed in each zone. The parameter *n* specifies the number of iterations per cycle (the default value is 5). In addition, *n* may have the following special values:

- 0 The indicated zone(s) will be bypassed, but the zone coupling will still take place. This is not a good idea as the active zone will update inactive ( $n = 0$ ) zones, overwriting the existing data. This inactive zone will then pass the active zone's data back to it.
- 1 Perform no iterations in the indicated zone and *do not* update this zone with adjacent zone information (but *do* pass this zone's frozen information to other zones each cycle).
- 2 Perform no iterations in the indicated zone and *do not* update this zone with adjacent zone information (and *do not* pass this zone's frozen information to other zones).

Residuals will be written to the *.lis* file every *freq* iterations, but only for cycles consistent with the PRINT FREQUENCY specified with the [CYCLES](#) keyword. The default value for *freq* is 1.

Note that this does not affect integrated properties specified using the [LOADS](#) keyword block. The print frequency for integrated properties is controlled by the PRINT keyword in the LOADS keyword block.

The *zone\_selector*, if specified, must appear last. The print frequency, however, is global; the value specified on the last ITERATIONS entry in the input data (*.dat*) file will be used in all zones.

#### Example

If the user specifies

```
CYCLES 1000 PRINT FREQUENCY 10
ITERATIONS PER CYCLE 10 PRINT FREQUENCY 5
```

cycle time information will be printed for cycles 1, 11, 21, etc. Residuals will be printed for iterations 10 (in cycle 1), 110 (cycle 11), 210 (cycle 21), etc.

See Also: [CYCLES](#), [LOADS](#), [NAVIER-STOKES ITERATIONS](#), [TURBULENCE](#)

**LOADS** — Flowfield integration (block)

```

LOADS
  [PRESSURE [OFFSET] {FREESTREAM | val}]
  PRINT [PLANES] [TOTALS] [ZONES] [LIFT | DRAG] [FREQUENCY freq] [MAKE_FRC file]
  [REFERENCE AREA aref]
  [REFERENCE LENGTH lref]
  [REFERENCE MOMENT CENTER xc yc zc]
  ZONE nzn
  SUBSET I range J range K range word1 [word2 [word3...]]
  SURFACE {I val | J val | K val} word1 [word2 [word3...]]
ENDLOADS

```

Flowfield properties may be integrated during the course of a WIND run to check convergence and solution quality. The values printed to the list file represent the force coefficients  $F/A_r q_r$  and moment coefficients  $M/L_r A_r q_r$  in the  $x$ ,  $y$ , and  $z$  coordinate directions, where  $F$  and  $M$  are the force and moment, and  $L_r$ ,  $A_r$ , and  $q_r$  are the reference values of length, area, and dynamic pressure. Integrated mass and momentum fluxes may also be computed. The values are written into the list output (.lis) file.

The various elements of the **LOADS** input block are defined as follows:

```
PRESSURE [OFFSET] {FREESTREAM | val}
```

This keyword controls the pressure integration. By default, the code uses  $P - P_\infty$  in all pressure integrations, where  $P_\infty$  is the static pressure at freestream conditions. This keyword permits the user to specify the pressure offset value, such that:

```

FREESTREAM    Use  $P - P_\infty$ 
val           Use  $P - val$ 

```

```
PRINT [PLANES] [TOTALS] [ZONES] [LIFT | DRAG] [FREQUENCY freq] [MAKE_FRC file]
```

This keyword controls the output data from the integration. Options are turned ‘on’ by including the appropriate keyword.

|                    |                                                                                                                                                                                                                                             |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>PLANES</b>      | Output the result of the integration for each subset specified.                                                                                                                                                                             |
| <b>TOTALS</b>      | Output integration grand totals over all the zones at the end of each cycle.                                                                                                                                                                |
| <b>ZONES</b>       | Output integration totals for each zone.                                                                                                                                                                                                    |
| <b>LIFT   DRAG</b> | Output lift, drag, and side forces instead of $(x, y, z)$ force components. Directions for the lift, drag, and side force components are computed from the angles of attack and yaw specified using the <a href="#">FREESTREAM</a> keyword. |
| <b>FREQUENCY</b>   | Output integration results (except for grand totals) every <i>freq</i> iterations.                                                                                                                                                          |
| <b>MAKE_FRC</b>    | Write grand totals to the summary file <i>file</i> at the end of each cycle, in addition to the .lis file. This requires that <b>PRINT TOTALS</b> also be specified.                                                                        |

|                         |                 |
|-------------------------|-----------------|
| REFERENCE AREA          | <i>aref</i>     |
| REFERENCE LENGTH        | <i>lref</i>     |
| REFERENCE MOMENT CENTER | <i>xc yc zc</i> |

These keywords specify the reference area *aref* in square inches; the reference length *lref* in inches; and the coordinates (*xc,yc,zc*) of the reference moment center in inches. The default values are 1.0 for *aref*, 1.0 for *lref*, and (0.0, 0.0, 0.0) for (*xc, yc, zc*).

|      |            |
|------|------------|
| ZONE | <i>nzn</i> |
|------|------------|

This keyword specifies the zone number *nzn* for the subsets which follow.

|        |   |              |   |              |   |              |              |                |                      |
|--------|---|--------------|---|--------------|---|--------------|--------------|----------------|----------------------|
| SUBSET | I | <i>range</i> | J | <i>range</i> | K | <i>range</i> | <i>word1</i> | [ <i>word2</i> | [ <i>word3...</i> ]] |
|--------|---|--------------|---|--------------|---|--------------|--------------|----------------|----------------------|

The *range* parameters define a subset of a computational surface in zone *nzn* over which the integration is to be performed, and take one of the following forms:

*index1 index2* Starting and ending indices in the specified direction. **LAST** may be used for the last index.

**ALL** Equivalent to 1 **LAST**.

For three-dimensional cases, the starting and ending indices for one (and only one) of the I, J, or K parameters must be the same. For two-dimensional cases, the K parameter must be specified as either K 1 1 or K **ALL**; one or both of the I and J parameters may have different starting and ending indices.

The *word* parameters are keywords specifying what flowfield properties are to be computed, plus a couple of additional control values. The following keywords are currently available:

|                            |                                                                                                                                                                                                                                                                    |
|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>FORCE</b>               | Compute pressure forces.                                                                                                                                                                                                                                           |
| <b>MASS</b>                | Compute mass flow.                                                                                                                                                                                                                                                 |
| <b>MOMENT</b>              | Compute moments.                                                                                                                                                                                                                                                   |
| <b>MOMENTUM</b>            | Compute momentum.                                                                                                                                                                                                                                                  |
| <b>VISCOUS</b>             | Compute viscous force and moments.                                                                                                                                                                                                                                 |
| <b>NORMAL</b> <i>inorm</i> | <i>inorm</i> = ±1, specifying the desired normal vector direction; +1 for the increasing index direction and -1 for the decreasing index direction. <i>inorm</i> defaults toward the interior of the grid if the integration surface is on a boundary of the zone. |
| <b>NOSLIP</b>              | Integrate only over points where the total velocity is 0.                                                                                                                                                                                                          |

|         |    |            |  |   |            |  |   |              |              |                |                      |
|---------|----|------------|--|---|------------|--|---|--------------|--------------|----------------|----------------------|
| SURFACE | {I | <i>val</i> |  | J | <i>val</i> |  | K | <i>val</i> } | <i>word1</i> | [ <i>word2</i> | [ <i>word3...</i> ]] |
|---------|----|------------|--|---|------------|--|---|--------------|--------------|----------------|----------------------|

This keyword is similar to the **SUBSET** keyword, except that it provides a slightly simpler syntax for specifying a complete coordinate surface in zone *nzn* over which the integration is to be performed. The parameter *val* is a coordinate index specifying the surface.

The word parameters are the same as those described above for the SUBSET keyword.

Example

```
LOADS
  PRESSURE OFFSET 0.0
  PRINT PLANES ZONES TOTALS LIFT FREQUENCY 2
  REFERENCE AREA 100.
  REFERENCE LENGTH 12.
  REFERENCE MOMENT CENTER 35. 36. 78.
  ZONE 4
    SUBSET I ALL J 1 1 K 15 LAST FORCE MOMENT VISCOUS NOSLIP
  ZONE 10
    SURFACE I 1 MOMENTUM MASS
ENDLOADS
```

**MARCHING** — Parabolized Navier-Stokes algorithm

|                                                                                                       |
|-------------------------------------------------------------------------------------------------------|
| <b>MARCHING</b> [ <b>LIMITER</b> <i>value</i> ] [ <b>CHECKPOINT</b> <i>interval</i> ] [ <b>COPY</b> ] |
|-------------------------------------------------------------------------------------------------------|

This keyword enables WIND's spatial marching, or parabolized Navier-Stokes (PNS), algorithm for flowfields which are supersonic in the computational  $i$ -direction. In this mode, WIND marches from the  $i = 1$  to the  $i = i_{max}$  computational surface, attempting to compute a steady-state solution at each plane before moving on to the next one. Using the PNS algorithm significantly reduces the computing time required for supersonic solutions.

**LIMITER** *value*                      This keyword enables the marching limiter, which limits the change in the solution vector  $\mathbf{Q}$  to  $(value)\mathbf{Q}$ . I.e.,

$$\Delta\mathbf{Q} \leq (value)\mathbf{Q}$$

**CHECKPOINT** *interval*              The parameter *interval* specifies the number of  $i$ -planes to be computed before writing the current flowfield to the solution file. The default is 10.

**COPY**                                      This keyword requests that WIND copy the solution from the most recently computed  $i$ -plane to the upcoming  $i$ -plane, giving a (hopefully) better initialization to the new  $i$ -plane than simply starting from freestream flow.

Notes

- Marching is only available when WIND is run in serial mode.
- Marching is not available with the one-equation turbulence models (**BARTH**, **SPALART**).
- The default explicit operator (**RHS PHYSICAL SECOND**) uses the  $i+1$  grid point, which is invalid in a PNS solution. WIND automatically changes the explicit operator to **RHS PHYSICAL FIRST** in the  $i$ -direction even if **RHS PHYSICAL SECOND** is specified.

See Also: [RHS](#), [TURBULENCE](#)

**MASS FLOW** — Outflow boundaries, mass flow

MASS [FLOW] {RATE [ACTUAL | CORRECTED] | RATIO} *value* [[ZONE] *zone*]

These keywords allow the user to specify mass flow at an outflow boundary in the flowfield. The specified *value* must be positive.

**RATE** *value* represents the mass flow rate in lb<sub>m</sub>/sec, and may be actual (the default) or corrected, as specified by the **ACTUAL** or **CORRECTED** keyword.

**RATIO** *value* represents the mass flow ratio. The actual mass flow is computed as

$$\dot{m} = (value)\rho_{\infty}U_{\infty}A_{cap}$$

where  $A_{cap}$  is the capture area found in the *.cgd* file zonal parameters.

**ZONE** *zone* is the zone number for application of the specified mass flow. The default zone number is 1.

The corrected air flow is defined as

$$W_c = W_{actual} \frac{\theta_x^{0.5}}{\delta_x}$$

where

$$\delta_x = P_x/P_0 \quad \theta_x = T_x/T_0$$

and  $P_x$  and  $T_x$  are the total pressure and temperature at the duct exit, and  $P_0$  and  $T_0$  are equal to 14.7 psi and 520 °R, respectively.

Examples

```
MASS FLOW RATIO          0.95 ZONE 2
MASS FLOW RATE  ACTUAL    180. ZONE 3
MASS FLOW RATE  CORRECTED 220. ZONE 4
```

See Also: [COMPRESSOR FACE](#), [DOWNSTREAM PRESSURE](#), [DOWNSTREAM MACH](#)

**MFD** — Magneto-Fluid Dynamics Model (block)

```

MFD
  [OUTPUT {BFIELD | CONDUCTIVITY | CURRENT | EFIELD | VOLTAGE | LORENTZ}]
  [RELAX_MFD nriter]
  [UPDATE nuit]
  [RADIATION emiss lref [tback]]
  {LORENTZ FILE fn FREQUENCY f [DUTY du] [SCALE sc] \
    PHASES n PATTERNS p1 p2 p3 ... pn
  |
  {CONDUCTIVITY | SIGMA} {CONSTANT sigma | \
    CFL | \
    EQUILIBRIUM {ARGON | AIR | GAS} [POTASSIUM mk] | \
    LINEAR t1 sig1 t2 sig2 | \
    PREDICTED [USING] [LIN-RESSLER | BOEING]}
  BFIELD {CONSTANT bz | BLOCKS nblocks | CFL}
  {EFIELD {CONSTANT ey | BLOCKS neblocks | CFL} | \
    VOLTAGE {BOUNDARIES nvbnd | CFL | PARAMETERS mitvlt vlttol vltrx vlttry vlttrz vltfac}
  }
}
ENDMFD

```

The MFD keyword block allows the user to specify the magneto-fluid dynamics mode and input data for the desired forms of the magnetic and electric fields, the electrical conductivity, output variables, and approximate radiation energy losses.

**Control Functions**

MFD

Defines the beginning of the MFD block.

ENDMFD

Defines the end of the MFD block.

OUTPUT {BFIELD | CONDUCTIVITY | CURRENT | EFIELD | VOLTAGE | LORENTZ}

The specified data will be written into the flow (*.cfl*) file. Multiple OUTPUT keywords may be specified, to write multiple types of data into the *.cfl* file.

RELAX\_MFD *nriter*

The MFD source terms will be relaxed over *nriter* iterations. The default for *nriter* is 1.

UPDATE *nuit*

The MFD source terms will be updated every *nuit* Navier-Stokes iterations. The default for *nuit* is 1.



**RADIATION** *emiss lref [tback]*

Estimate the energy loss due to thermal radiation of the fluid with emissivity *emiss*, optical depth *lref*, and background temperature *tback* (°R). The default for *tback* is the freestream static temperature.

### Body Force Determination

The body force resulting from the MFD terms can be added in one of two ways: (1) by directly specifying the Lorentz force; or (2) by specifying the conductivity, the magnetic field, and either the electric field or voltage.

The following keyword is used to directly specify the Lorentz force. If this method is used, the CONDUCTIVITY, BFIELD, EFIELD, and VOLTAGE keywords are not allowed.

**LORENTZ FILE** *fn* **FREQUENCY** *f* [**DUTY** *du*] [**SCALE** *sc*] \  
**PHASES** *n* **PATTERNS** *p1 p2 p3 ... pn*

*nl* Lorentz force distributions are stored in a *.cem* file. This distribution is spread over space depending upon the phasing patterns. The pattern changes *n* times during a cycle.

|                                        |                                                                                                                                     |
|----------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| <b>FILE</b> <i>fn</i>                  | Specifies the name of the <i>.cem</i> file containing the Lorentz force patterns                                                    |
| <b>FREQUENCY</b> <i>f</i>              | Number of cycles/second                                                                                                             |
| <b>DUTY</b> <i>du</i>                  | Force is on the first <i>du</i> portion of a phase. The default value is 1.0.                                                       |
| <b>SCALE</b> <i>sc</i>                 | Multiply the Lorentz force by <i>sc</i> before adding it to the equations. The default value is 1.0.                                |
| <b>PHASES</b> <i>n</i>                 | Number of pattern phases in a cycle                                                                                                 |
| <b>PATTERNS</b> <i>p1 p2 p3 ... pn</i> | The Lorentz force distribution to use in each phase. Must correspond to the <i>nl</i> distributions stored in the <i>.cem</i> file. |

If the Lorentz force is not directly specified using the **LORENTZ** keyword, you must use the **CONDUCTIVITY** and **BFIELD** keywords to specify the conductivity and the magnetic field, and either the **EFIELD** or **VOLTAGE** keywords to specify the electric field or voltage.

**{CONDUCTIVITY | SIGMA}** **{CONSTANT** *sigma* | \  
CFL | \  
EQUILIBRIUM {**ARGON** | **AIR** | **GAS**} [**POTASSIUM** *mk*] | \  
**LINEAR** *t1 sig1 t2 sig2* | \  
**PREDICTED** [**USING**] [**LIN-RESSLER** | **BOEING**]}

This keyword specifies the electrical conductivity in mhos/meter.

|                              |                                                          |
|------------------------------|----------------------------------------------------------|
| <b>CONSTANT</b> <i>sigma</i> | Hold the conductivity constant at the value <i>sigma</i> |
| <b>CFL</b>                   | Read the conductivity from the flow ( <i>.cfl</i> ) file |

EQUILIBRIUM {ARGON | AIR | GAS} [POTASSIUM *mk*]

Estimate the electron density as a function of temperature for the indicated gas as input to the Lin & Ressler conductivity model. The default gas is air. If POTASSIUM *mk* is specified, the effect of the mass fraction *mk* of potassium will be included.

LINEAR *t1 sig1 t2 sig2*

Set the conductivity to *sig1* at and below the temperature *t1*; to *sig2* at and above the temperature *t2*; and use a linear distribution for temperatures between *t1* and *t2*. The temperatures are in °R.

PREDICTED [USING] [LIN-RESSLER | BOEING]

Use real-gas predicted electron densities for input to the indicated conductivity model. The default is the Lin & Ressler model.

BFIELD {CONSTANT *bz* | BLOCKS *nbblocks* | CFL}

This keyword specifies the magnetic field in tesla.

CONSTANT *bz*

Hold the magnetic field constant at the value *bz*, in the *z* coordinate direction

BLOCKS *nbblocks*

Specify the magnetic field by reading in *nbblocks* blocks of data containing the magnetic field vector at selected coordinate points. The data immediately follows the BFIELD BLOCKS keyword. See the **Field Block Description** for details and an example.

CFL

Read the magnetic field from the flow (.cfl) file

EFIELD {CONSTANT *ey* | BLOCKS *neblocks* | CFL}

This keyword specifies the electric field in Volts/meter.

CONSTANT *ey*

Hold the electric field constant at the value *ey*, in the *y* coordinate direction

BLOCKS *neblocks*

Specify the electric field by reading in *neblocks* blocks of data containing the electric field vector at selected coordinate points. The data immediately follows the EFIELD BLOCKS keyword. See the **Field Block Description** for details and an example.

CFL

Read the electric field from the flow (.cfl) file

VOLTAGE {BOUNDARIES *nvbnd* | CFL | PARAMETERS *mitvlt vlttol vltvx vltvy vltvz vltfac*}

With this keyword the electric field is determined by specifying the electric potential.

BOUNDARIES *nvbnd*

Specify the electric potential at *nvbnd* zonal boundary regions. The data immediately follows the VOLTAGE BOUNDARIES keyword. See the **Voltage Boundary Description** for details and an example.

CFL

Read the electric potential field from the flow (.cfl) file

PARAMETERS *mitvlt vlttol vltrx vltry vltrz vltfac*

Iterate a maximum of *mitvlt* iterations (the default is 10,000) to a tolerance of *vlttol* (a positive value means to a level of *vlttol*, a negative value means  $|vlttol|$  orders of magnitude; the default is a level of  $10^{-12}$ ) with implicit factors *vltrx*, *vltry*, and *vltrz* (currently these must all be set to 0.0) and with an over-relaxation factor of *vltfac* (a typical value is 0.4).

### Field Block Description

By using the **BFIELD BLOCKS** and/or **EFIELD BLOCKS** keywords, the magnetic and/or electric field may be determined from blocks containing the field vector at selected coordinate points. Each block consists of eight points in space, with the corresponding field vector at each of those points. The region contained within the blocks is filled using tri-linear interpolation between the specified points. Up to eight blocks may be specified for each field type. The spatial locations of the blocks may overlap, with the later-specified blocks overwriting the earlier ones.

There are eight lines of input per block, one for each of the eight coordinate points. Each line contains six values—the *x*, *y*, and *z* coordinates, and the field vector components in the *x*, *y*, and *z* directions.

The following example specifies the magnetic field using two blocks. (The comments in a slanted font are not part of the input.)

|                 |          |          |                      |                      |                      |                                    |
|-----------------|----------|----------|----------------------|----------------------|----------------------|------------------------------------|
| BFIELD BLOCKS 2 |          |          |                      |                      |                      |                                    |
| 60.0            | 0.0      | 0.0      | 0.0                  | 0.0                  | 0.0                  | <i>Upstream plane of block 1</i>   |
| 60.0            | 20.0     | 0.0      | 0.0                  | 0.0                  | 0.0                  |                                    |
| 60.0            | 0.0      | 1.0      | 0.0                  | 0.0                  | 0.0                  |                                    |
| 60.0            | 20.0     | 1.0      | 0.0                  | 0.0                  | 0.0                  |                                    |
| 120.0           | 0.0      | 0.0      | 0.0                  | 0.0                  | 10.0                 | <i>Downstream plane of block 1</i> |
| 120.0           | 20.0     | 0.0      | 0.0                  | 0.0                  | 5.0                  |                                    |
| 120.0           | 0.0      | 1.0      | 0.0                  | 0.0                  | 10.0                 |                                    |
| 120.0           | 20.0     | 1.0      | 0.0                  | 0.0                  | 5.0                  |                                    |
| 120.0           | 0.0      | 0.0      | 0.0                  | 0.0                  | 10.0                 | <i>Upstream plane of block 2</i>   |
| 120.0           | 20.0     | 0.0      | 0.0                  | 0.0                  | 5.0                  | <i>(same as downstream of 1)</i>   |
| 120.0           | 0.0      | 1.0      | 0.0                  | 0.0                  | 10.0                 |                                    |
| 120.0           | 20.0     | 1.0      | 0.0                  | 0.0                  | 5.0                  |                                    |
| 180.0           | 0.0      | 0.0      | 0.0                  | 0.0                  | 0.0                  | <i>Downstream plane of block 2</i> |
| 180.0           | 20.0     | 0.0      | 0.0                  | 0.0                  | 0.0                  |                                    |
| 180.0           | 0.0      | 1.0      | 0.0                  | 0.0                  | 0.0                  |                                    |
| 180.0           | 20.0     | 1.0      | 0.0                  | 0.0                  | 0.0                  |                                    |
| <i>x</i>        | <i>y</i> | <i>z</i> | <i>B<sub>x</sub></i> | <i>B<sub>y</sub></i> | <i>B<sub>z</sub></i> |                                    |

### Voltage Boundary Description

Voltage boundaries are used to specify the boundary conditions at zone boundaries for the electric potential solver. The electric potential solver will take into account gradients and discontinuities in conductivity, as well as the electromotive force (EMF) induced by the fluid motion through the magnetic field. Currently, accurate voltage predictions require a grid with low skewness.

Up to 64 boundaries may be specified. The conditions at each boundary are specified on a single line, with eight values—*zone*, *face*, *type*, *L1*, *L2*, *M1*, *M2*, *value*—defined as follows:

*zone*                      Zone containing the boundary

| <i>face</i>           | The boundary face, specified as a number from 1 to 6 corresponding to the $i_1$ , $i_{max}$ , $j_1$ , $j_{max}$ , $k_1$ , and $k_{max}$ face, respectively                                                                                                                                                                                                                                                                                                                                                                                                    |             |                |                    |                                                 |                    |                                                 |                    |                                                 |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|----------------|--------------------|-------------------------------------------------|--------------------|-------------------------------------------------|--------------------|-------------------------------------------------|
| <i>type</i>           | The boundary type, specified as 0 for a reflection boundary, and 1 for specified voltage                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |             |                |                    |                                                 |                    |                                                 |                    |                                                 |
| <i>L1, L2, M1, M2</i> | The indices on the face over which the boundary condition applies, as follows                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |             |                |                    |                                                 |                    |                                                 |                    |                                                 |
|                       | <table> <tr> <th><u>Face</u></th><th><u>Indices</u></th></tr> <tr> <td><math>i_1</math> or <math>i_{max}</math></td><td><math>j_{low}</math>, <math>j_{high}</math>, <math>k_{low}</math>, <math>k_{high}</math></td></tr> <tr> <td><math>j_1</math> or <math>j_{max}</math></td><td><math>k_{low}</math>, <math>k_{high}</math>, <math>i_{low}</math>, <math>i_{high}</math></td></tr> <tr> <td><math>k_1</math> or <math>k_{max}</math></td><td><math>i_{low}</math>, <math>i_{high}</math>, <math>j_{low}</math>, <math>j_{high}</math></td></tr> </table> | <u>Face</u> | <u>Indices</u> | $i_1$ or $i_{max}$ | $j_{low}$ , $j_{high}$ , $k_{low}$ , $k_{high}$ | $j_1$ or $j_{max}$ | $k_{low}$ , $k_{high}$ , $i_{low}$ , $i_{high}$ | $k_1$ or $k_{max}$ | $i_{low}$ , $i_{high}$ , $j_{low}$ , $j_{high}$ |
| <u>Face</u>           | <u>Indices</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |             |                |                    |                                                 |                    |                                                 |                    |                                                 |
| $i_1$ or $i_{max}$    | $j_{low}$ , $j_{high}$ , $k_{low}$ , $k_{high}$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |             |                |                    |                                                 |                    |                                                 |                    |                                                 |
| $j_1$ or $j_{max}$    | $k_{low}$ , $k_{high}$ , $i_{low}$ , $i_{high}$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |             |                |                    |                                                 |                    |                                                 |                    |                                                 |
| $k_1$ or $k_{max}$    | $i_{low}$ , $i_{high}$ , $j_{low}$ , $j_{high}$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |             |                |                    |                                                 |                    |                                                 |                    |                                                 |
| <i>value</i>          | The voltage for specified-voltage boundaries; 0.0 for reflection boundaries                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |             |                |                    |                                                 |                    |                                                 |                    |                                                 |

The following example specifies eight voltage boundaries. (The comments in a slanted font are not part of the input.)

```

VOLTAGE BOUNDARIES 8
  1    1    0    1   30    1    1    0.0
  1    1    1   31   61    1    1    0.0
  1    2    1    1   31    1    1  10000.0
  1    2    0   32   61    1    1    0.0
  1    3    0    1    1    1   30    0.0
  1    3    1    1    1   31   61  10000.0
  1    4    1    1    1    1   31    0.0
  1    4    0    1    1   32   61    0.0

Zone Face Type  L1  L2  M1  M2  Value

```

**MOVING WALL** — Specify moving wall boundaries

|                                                                                               |
|-----------------------------------------------------------------------------------------------|
| MOVING [WALL] <i>ibldreg</i> {[TRANSLATE {I   X} <i>speed</i> ]   [SPIN X <i>rps yc zc</i> ]} |
|-----------------------------------------------------------------------------------------------|

This keyword allows no-slip walls to have a tangential velocity.

*ibldreg*      Moving walls are activated in regions specified as bleed in the *.cgl* file (set with GMAN). *ibldreg* is the bleed region number.

TRANSLATE I    Move wall in the *i*-direction at the velocity *speed*, in feet per second.

TRANSLATE X    Move wall in the *x*-direction at the velocity *speed*, in feet per second.

SPIN X          Move wall to model rotation about the *x*-axis (wall should be spinning and tangent to this motion).

*rps*          Rotation speed in radians per second

*yc,zc*        Coordinates of the center of rotation in the *y-z* plane

**NAVIER-STOKES ITERATIONS** — Navier-Stokes sub-iterations

`NAVIER-STOKES ITERATIONS iter [zone_selector]`

The WIND code organizes the equations to be solved into logical “groups” that are solved together. It also allows multiple iterations of a specific group (i.e., sub-iterations) for each “iteration per cycle”. (The number of iterations per cycle is set using the [ITERATIONS](#) keyword.)

The **NAVIER-STOKES ITERATIONS** keyword allows the user to specify the number of sub-iterations for the Navier-Stokes equation group (which includes any chemical species equations) performed in each zone for each “iteration per cycle”. The default value is one, indicating that each “iteration per cycle” corresponds to one iteration of the Navier-Stokes equations. Note that if *iter* is set to zero, the Navier-Stokes equations will not be solved at all.

*See Also:* [ITERATIONS](#), [TURBULENCE](#)

**NEWTON** — Use Global Newton time stepping

**NEWTON** [TIME LEVELS *newstp*] [CONVERGE {LEVEL | ORDER} *newcvg*]

This keyword specifies that the Global Newton time stepping algorithm is to be used. The parameter *newstp* specifies the number of Global Newton time levels to advance. The default value is 30.

There are two possible procedures for determining the overall global convergence.

**CONVERGE LEVEL** Convergence is assumed when

$$|\mathbf{Q}^n - \mathbf{Q}^{n-1}| < \text{newcvg}$$

where  $\mathbf{Q}$  represents the vector of dependent variables, and  $n$  is the Newton time level.

**CONVERGE ORDER** Convergence is assumed when

$$\frac{|\mathbf{Q}^n - \mathbf{Q}^{n-1}|}{\max(|\mathbf{Q}^n - \mathbf{Q}^{n-1}|)} < 10^{-\text{newcvg}}$$

The default is **CONVERGE ORDER 3**.

Within a Newton time level, the number of cycles and iterations to be run is specified using the **CYCLES** and **ITERATIONS** keywords.

The convergence criteria within a Newton time level may be specified using the **CONVERGE** keyword. Note that the default for the **CONVERGE** keyword is a four-order of magnitude decrease in the maximum residual. If Newton iteration is being used for a steady flow problem, with the default of three orders of magnitude for the global convergence criteria, it would make sense to also use three orders of magnitude for the convergence criteria within a Newton time step.

*Note: This keyword does not currently work when **assignment mode shared** is specified in the multi-processing control (.mpc) file. See [Section 9.1](#).*

See Also: **CYCLES**, **ITERATIONS**, **CONVERGE**

**OUTFLOW NON-REFLECTING** — Outflow boundaries, non-reflecting

**OUTFLOW NON-REFLECTING** *zone*

This keyword imposes a non-reflecting, subsonic outflow boundary condition at outflow boundaries. Acoustic disturbances reaching the boundary are essentially eliminated. It is actually implemented using the Paynter compressor face model ([Slater and Paynter, 2000](#)), but with the response coefficients  $\alpha$  and  $\beta$  set to zero.

This boundary condition is intended to be used for time-accurate flows with a CFL number less than one; however, it may be applicable in certain cases for steady-state simulations.

*See Also:* [COMPRESSOR FACE PAYNTER](#)



**PERIODIC** — Periodic boundaries

Periodic boundaries are treated as normal coupled boundaries in WIND, with the connection data stored in the grid (*.cgd*) file. No keyword is needed in the input data (*.dat*) file.

GMAN's **CONNECT MODE** option is used to specify how to move one boundary surface to overlay it on the corresponding periodic boundary surface. There are two connection modes — translation and rotation. For translation, the user defines the  $\Delta x$ ,  $\Delta y$ , and  $\Delta z$  movement for the surface. For rotation the user sets the center of rotation and the angles. The periodic boundary surfaces do not have to be in the same zone, nor do they have to be point matched. They only have to line up physically once the movement has been performed.

Details about how to set a periodic boundary condition in GMAN are presented in [Section 5.6.1](#) starting on page 48.

**REINITIALIZE** — Reinitialize selected flowfield zones on restart

```
REINITIALIZE {MODE {AUTO | NOAUTO} | zone_selector [IJK_RANGE]}
```

This keyword may be used to reinitialize the flow conditions after a restart. The reinitialization will be done as if WIND were being run from scratch, i.e., to freestream values, to values specified using the [ARBITRARY INFLOW](#) keyword, and/or to values specified using the [BL\\_INIT](#) keyword, as described in [Section 3.5](#) starting on p. 35.

REINITIALIZE may be used in conjunction with the [ARBITRARY INFLOW](#) keyword to reinitialize only selected portions of the flow. If the IJK\_RANGE parameter is specified with the REINITIALIZE keyword, conditions will be reinitialized only at those points specified via the IJK\_RANGE parameter on the ARBITRARY INFLOW keyword. If the IJK\_RANGE parameter is not specified with the REINITIALIZE keyword, the conditions at all the grid points in the specified zones will be reinitialized.

Multiple REINITIALIZE keywords are permitted, each on a separate line in the input data file.

If the MODE NOAUTO option is used (the default), and if the grid sizes in the grid (.cgd) and flow (.cfl) files are different, WIND will stop unless: (1) [ITERATIONS PER CYCLE](#) was set to -2 for the zone in question, indicating that the zone plays no part in the calculation; or (2) the user specified that the zone be reinitialized, using REINITIALIZE *zone\_selector*.

If the MODE AUTO option is used, and if the grid sizes in the .cgd and .cfl files are different, WIND will automatically reset the zone size in the .cfl file to match the size in the .cgd file, and reinitialize the zone.

See Also: [ARBITRARY INFLOW](#), [BL\\_INIT](#), [FREESTREAM](#)

**RELAX COUPLING** — Set zone coupling relaxation factor

```
RELAX {COUPLE | COUPLING} factor [ZONE nzn] \
      [BOUNDARY] [ALL | I1 | IMAX | J1 | JMAX | K1 | KMAX]
```

This keyword allows the user to specify the relaxation factor, when using characteristic zone coupling.

*factor*    Relaxation factor (0.0 for no coupling, 1.0 for full zone coupling)

*nzn*        Zone number. Omit or set to 0 to apply *factor* in all zones.

If this keyword is not used, a relaxation factor of 0.7 will be applied for steady-state calculations. For space-marching and time-accurate calculations, the relaxation factor is automatically set to 1.0.

**REL-ROT-ZONE** — Relative rotating zones (block)

```

REL-ROT-ZONE
  ZONE iz1 BOUNDARY {I1 | IMAX | J1 | JMAX | K1 | KMAX} \
    [SUBSET I range J range K range]
  ZONE iz2 BOUNDARY {I1 | IMAX | J1 | JMAX | K1 | KMAX} \
    [SUBSET I range J range K range]
  ROTATING-ZONE flag
ENDRRZ

```

The **REL-ROT-ZONE** keyword block, along with the **ROTATE** keyword, may be used to specify that one zone is rotating relative to another zone. This “relative rotating zone” capability is intended to simulate rotating devices such as compressor fans. The **ROTATE** keyword is used to specify which zone(s) are rotating, plus the rotation axis and rate. The **REL-ROT-ZONE** keyword block specifies the location of the interface between the two zones, and how flow conditions are to be transferred between zones.

Zones sharing an interface may have different circumferential extents. Thus, when modeling a turbomachinery component like a compressor, only one blade per stage is required. Multiple zones and zonal interfaces may be used to cover the radial extent of the stage-to-stage interface, but a single zone must be used in the circumferential direction. The interface surface must correspond to a surface of revolution and grid lines in the circumferential direction must be at a constant radius relative to the rotation axis. The rotation axis must correspond to a coordinate axis.

As noted above, this capability is intended to simulate rotating devices such as compressor fans. A typical configuration would be an upstream non-rotating zone covering the full  $360^\circ$  cross-section, and a rotating downstream zone (or zones, if multiple zones are used in the radial direction) with a circumferential extent of  $360^\circ/N$  corresponding to a single blade. Periodic boundary conditions would be set at the circumferential boundaries in the rotating zone(s), using GMAN’s rotational coupling mode, as described in [Section 5.6.1](#).

The coupling of the downstream face of the non-rotating zone to the upstream face of the rotating zone would also be done using GMAN’s rotational coupling mode, repeated  $N - 1$  times. This will couple all the points except for that portion of the face that corresponds to the downstream zone in its non-rotated position. These remaining points are then coupled using ordinary (i.e., non-rotated) coupling mode.<sup>26</sup>

The elements of the **REL-ROT-ZONE** keyword block are defined as follows:

```
REL-ROT-ZONE
```

Defines the beginning of the relative rotating zone block.

<sup>26</sup> In WIND, non-zero rotation angles trigger the use of a rotationally periodic boundary condition. Since this is *not* what we want between two relative rotating zones, this “non-rotated” coupling should be done *last*, so that zero rotation angles are written into the common grid (.cgd) file.

```

ZONE iz1 BOUNDARY {I1 | IMAX | J1 | JMAX | K1 | KMAX} \
      [SUBSET I range J range K range]
ZONE iz2 BOUNDARY {I1 | IMAX | J1 | JMAX | K1 | KMAX} \
      [SUBSET I range J range K range]

```

These two lines define the interface between the two zones. The relevant zones are given by the values of *iz1* and *iz2*, and the relevant boundaries within zones *iz1* and *iz2* are specified via the BOUNDARY keyword parameter.

*iz1*     Zone to which increments will be added when passing info to *iz2*

*iz2*     Zone receiving positive increments, increments will be subtracted when passing information back to zone *iz1*

The SUBSET parameter may be used to specify that the change in properties occurs only over a part of the zone boundary. Otherwise, it is assumed that the change occurs over the entire boundary. The *range* parameters define the part of the zone boundary over which the change occurs, and take one of the following forms:

*index1 index2*     Starting and ending indices in the specified direction. LAST may be used for the last index.

ALL                 Equivalent to 1 LAST.

The starting and ending indices for the appropriate I, J, or K parameter (depending on the boundary specified) must be the same, and correspond to that boundary.

ROTATING-ZONE *flag*

*flag* defines how data is to be communicated between the two zones. Currently the only valid values are:

0     if the zones are not rotating relative to each other. This is the default.

2     for circumferential averaging. This permits the communication of the bulk fluid properties between zones, while maintaining radial distributions and the efficiency of local time stepping.

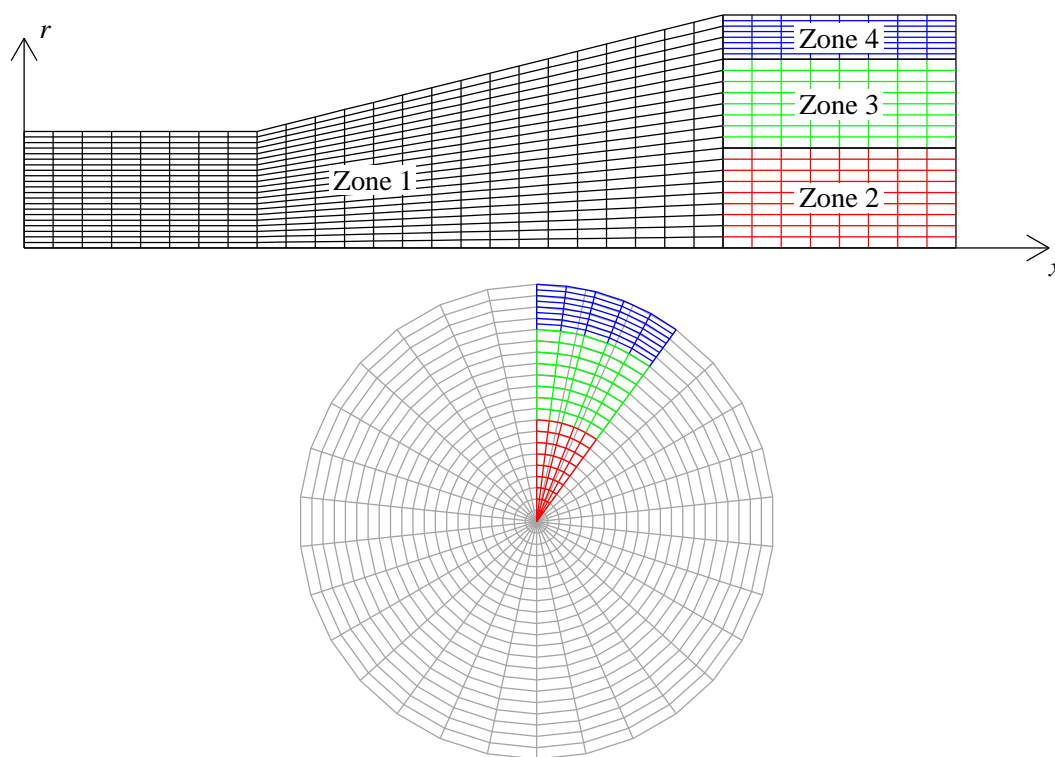
ENDRRZ

Defines the end of the relative rotating zone block.

### Example

Figure 8 shows a four-zone configuration. Zone 1 is non-rotating, and zones 2, 3, and 4 are rotating about the *x* axis at 1680 radians/sec in the counter-clockwise direction. The figure shows the grid from the side in a constant- $\theta$  plane, and at the interface plane between the non-rotating and rotating zones, looking downstream.

The non-rotating zone 1 covers the full 360° cross section, but the rotating zones 2–4 cover just 36°. As noted previously, zones sharing an interface may have different circumferential extents.



**Figure 8:** Example grid with relative rotating zones

This particular configuration is similar to one that might be used to model a single blade from a compressor blade row with 10 blades.

The indices  $i$ ,  $j$ , and  $k$  are in the axial, radial, and circumferential directions, respectively.

For this configuration, the zone coupling was done in GMAN as described below. In this discussion, the “boundary zone” is the one containing grid points for which a boundary condition is being set. GMAN is used to set the connectivity between points in the boundary zone to points in the “source zone”. The source zone is specified using GMAN’s “SEL OTHER BND” menu choice.

1. Run “auto-couple”, which results in the following coupling:

| <u>Zone</u> | <u>Boundary</u> |   | <u>Zone</u> | <u>Boundary</u> |
|-------------|-----------------|---|-------------|-----------------|
| 1           | IMAX            | ← | 2           | I1              |
| 1           | IMAX            | ← | 3           | I1              |
| 1           | IMAX            | ← | 4           | I1              |
| 2           | JMAX            | ↔ | 3           | J1              |
| 3           | JMAX            | ↔ | 4           | J1              |

Note that “auto-couple” couples the upstream faces of zones 2–4 to the downstream face of zone 1, but not vice-versa.

2. Manually couple the zone 1 K1 surface ( $\theta = 0^\circ$ ) to the zone 1 KMAX surface ( $\theta = 360^\circ$ ), and vice versa, using ordinary coupling.
3. Couple the K1 and KMAX surfaces in zone 2 as periodic boundaries, as described in [Section 5.6.1](#).

- (a) Use zone 2, boundary K1, as the “boundary zone”, and zone 2, boundary KMAX, as the “source zone”.
  - (b) Use 1 (rotation) as the connection mode, a point on the  $x$  axis as the center of rotation, and 36.0 0.0 0.0 for the rotation angles (in degrees, without commas, and positive for the +K direction).
  - (c) Repeat the above two steps, but using zone 2, boundary K1, as the “boundary zone”, zone 2, boundary KMAX, as the “source zone”, and -36.0 0.0 0.0 for the rotation angles.
  - (d) Repeat the above three steps for zones 3 and 4.
4. Couple the zone 1 IMAX surface to the zone 2 I1 surface, using rotational coupling mode repeatedly. The procedure is very similar to specifying periodic coupling.
  - (a) For the first rotation:
    - In graphics mode, from GMAN’s main menu select BOUNDARY COND.
    - Pick zone 1, boundary IMAX (the “boundary zone”).
    - Select MODIFY BNDY.
    - Select COUPLE.
    - Select SEL OTHER BND.
    - Pick zone 2, boundary I1 (the “source zone”).
    - Select SET COUP MODE.
    - Click on the choice under CONNECT MODE (initially \*\* NONE \*\*).
    - Respond to the prompts at the bottom of the screen, using 1 (rotation) as the connection mode, a point on the  $x$  axis as the center of rotation, and 36.0 0.0 0.0 for the rotation angles.
    - Select CONNECT.
  - (b) For the next rotation:
    - Select COUPLE.
    - Select SEL OTHER BND.
    - Pick zone 2, boundary I1 (the “source zone”).
    - Select SET COUP MODE.
    - Click on the choice under CONNECT MODE (now set to ROTATION).
    - Respond to the prompts at the bottom of the screen as before, except using 72.0 0.0 0.0 for the rotation angles.
    - Select CONNECT.
  - (c) Repeat the previous step for rotation angles of 108°, 144°, 180°, 216°, 252°, 288°, and 324°.
5. Couple the zone 1 IMAX surface to the zone 2 I1 surface in its non-rotated position.
  - Select BOUNDARY COND.
  - Pick zone 1, boundary IMAX (the “boundary zone”).
  - Select MODIFY BNDY.
  - Select COUPLE.
  - Select SEL OTHER BND.
  - Pick zone 2, boundary I1 (the “source zone”).
  - Select COUPLE.

6. Repeat steps 4 and 5 twice, coupling the zone 1 IMAX surface to the I1 surfaces in zones 3 and 4.
7. Whew!

The relevant keyword input in the input data (*.dat*) file would be

```
ROTATE 0.0 0.0 0.0 -1680.0 0.0 0.0 ZONE 2:4

REL-ROT-ZONE
  ZONE 1 BOUNDARY IMAX SUBSET I LAST LAST J 1 18 K ALL
  ZONE 2 BOUNDARY I1
  ROTATING-ZONE 2
ENDRRZ
REL-ROT-ZONE
  ZONE 1 BOUNDARY IMAX SUBSET I LAST LAST J 19 34 K ALL
  ZONE 3 BOUNDARY I1
  ROTATING-ZONE 2
ENDRRZ
REL-ROT-ZONE
  ZONE 1 BOUNDARY IMAX SUBSET I LAST LAST J 35 LAST K ALL
  ZONE 4 BOUNDARY I1
  ROTATING-ZONE 2
ENDRRZ
```

*See Also:* [ROTATE](#)



**RESTART | START** — Begin run in specified zone

**{START | RESTART}** [*zone* [*plane*]]

This keyword allows the user to specify the zone (and plane, for a marching solution) in which the solution will start (or restart) when running in single-processor mode. In parallel mode, the starting zone number is automatically set to 1.

In both single-processor and parallel modes, if this keyword is not used, WIND checks for the existence of a flow (*.cfl*) file. If one exists, the solution will be restarted using the values in the *.cfl* file as initial conditions. If a *.cfl* file does not exist, WIND will create and initialize one.

- zone*     First zone entered. If not specified, the solution will begin in the zone after the last successfully completed zone. However, if [Global Newton iteration](#) is being used, the solution must start in zone 1, and specification of a *zone* value other than 1 is not allowed.
- plane*    *i*-plane to start/restart solution when running in [PNS marching mode](#). If omitted, WIND will automatically start at the next *i*-plane after the last one completed.

**RHS** — Explicit operator control

**RHS** *key1* *key2* [*zone\_selector*]

This keyword allows control of the explicit operator used within each zone. The parameters *key1* and *key2* are keywords specifying the type and order of the operator. The following options are currently available for *key1*:

| <u>Keyword</u> | <u>Result</u>                           |
|----------------|-----------------------------------------|
| CENTRAL        | Central difference                      |
| COAKLEY        | Coakley upwind                          |
| ROE            | Roe upwind                              |
| VAN [LEER]     | Van Leer upwind                         |
| PHYSICAL       | Roe upwind modified for stretched grids |

The following options are available for *key2*:

| <u>Keyword</u> | <u>Result</u>                                                 |
|----------------|---------------------------------------------------------------|
| FIRST          | First-order (ROE, VAN LEER, PHYSICAL, and COAKLEY only)       |
| SECOND         | Second-order                                                  |
| THIRD          | Third-order (ROE, VAN LEER, and COAKLEY only)                 |
| CENTRAL        | Central difference (ROE and VAN LEER only)                    |
| MIXED          | Mixed fully upwind and central scheme (ROE and VAN LEER only) |
| THIRDU         | Third-order fully upwind (ROE and VAN LEER only)              |
| FOURTHU        | Fourth-order upwind-biased (ROE and VAN LEER only)            |
| FOURTHC        | Fourth-order central (ROE and VAN LEER only)                  |
| FIFTH          | Fifth-order upwind-biased (ROE and VAN LEER only)             |

The default is **RHS PHYSICAL SECOND**, the second-order Roe upwind operator with modifications for stretched grids.

Note that the default zonal coupling procedure (high-order Roe) is only allowed for some of the above options. See the [COUPLING](#) keyword for details.

The Van Leer scheme cannot currently be used for multi-species flows.

See Also: [COUPLING](#)

**ROLL** — Specify roll about one of the coordinate axes

**ROLL** {X | Y | Z} *rat* *x1* *x2*

The **ROLL** keyword allows the user to specify roll about one of the coordinate axes.

*rat*      Rotation rate in radians per second

*x1*, *x2*      Coordinates of the center of rotation. The coordinates specified by *x1* and *x2* depend on the rotation axis, as shown in the following table:

| <u>Rotation axis</u> | <u><i>x1</i></u> | <u><i>x2</i></u> |
|----------------------|------------------|------------------|
| X                    | <i>y</i>         | <i>z</i>         |
| Y                    | <i>z</i>         | <i>x</i>         |
| Z                    | <i>x</i>         | <i>y</i>         |

Note that in this mode, the calculation is done in inertial space, and the grid is assumed to roll through inertial space at the specified rate. The grid velocity imparts the rotating component. Thus, in order to show the velocity relative to the rotating axes, the roll component of velocity must be subtracted (via the **CFPOST** [post-processing package](#) or another mechanism).

Note also that, unfortunately, unless **ROLL** is used in a true time-accurate mode which moves the grid each time step (not currently available), as well as giving it velocity, the **ROLL** capability neglects the rotational potential energy, and so is not correct for high rotation rates.

With this keyword, the velocity vector must be aligned with the coordinate axis. (To get an angle of attack, rotate the grid in **GMAN**.)

See Also: [ROTATE](#)

**ROTATE** — Perform calculation in a rotating frame of reference

**ROTATE**  $x_{cntr}$   $y_{cntr}$   $z_{cntr}$   $\Omega_x$   $\Omega_y$   $\Omega_z$

The **ROTATE** keyword allows the user to compute the flow in a rotating reference frame. The center of rotation and rotation rates are specified via the arguments:

$x_{cntr}, y_{cntr}, z_{cntr}$       Center of rotation, in grid units

$\Omega_x, \Omega_y, \Omega_z$               Rotation rates about the three Cartesian coordinate axes, in radians/sec

Note: A similar capability is given by the **ROLL** keyword, in which the grid is rotated about an axis. However, **ROLL** does the calculation in inertial space, and uses the grid velocity to impart a rotating component. Unfortunately, unless **ROLL** is run in a true time-accurate mode which moves the grid each time step (not currently available), as well as giving it velocity, the **ROLL** capability neglects the rotational potential energy, and so is not correct for high rotation rates.

*See Also:* [REL-ROT-ZONE](#), [ROLL](#), [ARBITRARY INFLOW](#)

**SEQUENCE** — Grid sequencing control

|                                                             |
|-------------------------------------------------------------|
| <b>SEQUENCE</b> <i>nsi nsj nsk</i> [ <i>zone_selector</i> ] |
|-------------------------------------------------------------|

This option allows the user to, in effect, thin the computational grid, thus reducing CPU requirements. It can be used to obtain quick starting solutions for the fine grid. Separate keywords may be used for each zone.

*nsi*    Number of sequencing levels in the *i* direction

*nsj*    Number of sequencing levels in the *j* direction

*nsk*    Number of sequencing levels in the *k* direction

For one sequencing level, every other grid point is used in the calculation. For two sequencing levels, the sequencing process is repeated, resulting in only every fourth grid point being used. And so on. At the end of the run, the solution is interpolated back onto the entire grid to aid in restarting the solution and to provide a continuous field for post-processing.

The number of grid points in the sequencing direction must be equal to  $2^n m + 1$ , where  $m$  is an integer and  $n$  is the number of sequencing levels. Thus for 1 level of sequencing, the grid must have an odd number of points in the sequencing direction. For two levels there must be  $4m + 1$  points, etc.

**SMOOTHING** — Add dissipation to explicit operator

```
SMOOTHING [SECOND val1] [FOURTH val2] [SMLIMIT val3] [zone_selector]
```

This keyword is used to apply second- and/or fourth-order dissipation to the explicit operator. The parameters *val1*, *val2*, and *val3* are the second-order smoothing coefficient, the fourth-order smoothing coefficient, and the smoothing limiter, respectively.

| <u>Parameter</u> | <u>Default</u> | <u>“TVD-Like” Value</u> |
|------------------|----------------|-------------------------|
| <i>val1</i>      | 1/4            | < 1                     |
| <i>val2</i>      | 1/256          | < 1/32                  |
| <i>val3</i>      | 1/4            | 1                       |

The default values shown in the above table represent a Jameson-type smoothing, and are good for an Euler analysis. For viscous problems, it is recommended that [TEST 49](#) be set equal to 1, and that *val1*, *val2*, and *val3* be set equal to 1, 1/64, and 2, respectively.

**SPAWN** — Run an external process from WIND

SPAWN "*process*" [FREQUENCY *freq*] [NOCHECKPOINT]

This keyword allows the user to start (spawn) a process without stopping WIND. The process can be any valid Unix script or executable. When the new process is spawned, WIND will wait until the spawned process completes before continuing.

WIND will tack the current cycle number onto the end of the *process* call for potential use by the process as a parameter.

The spawned process will be run in the same directory as WIND. See the description of the **-runinplace** and **-runroot** options to the *wind* script in [Section 8.1](#) for details about the WIND run directory.

The keyword arguments and options are:

|                |                                                                                                                                                                                                 |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>process</i> | Full path name to the process (must be in quotes, 80 characters max)                                                                                                                            |
| <i>freq</i>    | Spawn process every <i>freq</i> cycles (default is 1)                                                                                                                                           |
| NOCHECKPOINT   | By default, WIND will override the checkpoint interval so that it aligns with the spawn frequency. NOCHECKPOINT will disable the realignment and cause the checkpoint interval to be unchanged. |

Note that while the *process* is being executed, WIND's input and output files (i.e., the *.lis* file, *.cfl* file, etc.) remain connected to the Fortran units used by WIND. Thus, if the *process* runs a program like *resplt* or *cfpost*, these files cannot be used directly as input. Instead, the *process* must copy the needed file(s) to temporary files, and use the temporary files as input.

#### Example

A common use for the SPAWN keyword, especially for unsteady flows, is to save intermediate *.cfl* files during a WIND run. A shell script may be SPAWN'ed at specified intervals to save the file under a unique name.

For example, consider the following shell script, named *save\_cfl*.

```
#!/bin/sh
#
# save_cfl - Save intermediate WIND .cfl files. This script is intended
#           to be spawned during a WIND run using the SPAWN keyword, as
#
#           SPAWN "path-to-script path-to-cfl" FREQUENCY num
#
#           where 'path-to-script' is the full path name for this
#           script, 'path-to-cfl' is the full path name for the saved
#           .cfl files (the current cycle number will automatically
#           be appended to this name), and 'num' is the frequency in
#           cycles for the SPAWN command. Note that the quote marks
#           around "path-to-script path-to-cfl" must be included.
```

```
# Check for enough arguments

if [ $# -lt 2 ]
then
    echo 'Error: save_cfl: Path name for saved .cfl files not specified' >& 2
    echo '      Use WIND's SPAWN keyword, as:' >& 2
    echo '      SPAWN "path-to-script path-to-cfl" FREQUENCY num' >& 2
    exit 1
fi

# $1 is path name for the saved .cfl files, $2 is current cycle number

cp FOR020 $1.$2
```

If the above script is located in the directory */home/user/bin*, the SPAWN keyword may be used as follows to save the *.cfl* file every 1000 cycles in the directory */home/user/wind*.

```
SPAWN "/home/user/bin/save_cfl /home/user/wind/runa.cfl" FREQUENCY 1000
```

Since the cycle number is automatically added to the end of the *process* when it's spawned, the above use of the SPAWN keyword causes the *save\_cfl* script to be run as

```
/home/user/bin/save_cfl /home/user/wind/runa.cfl ncycle
```

where *ncycle* is the current cycle number. The result is that the intermediate *.cfl* files will be stored in the directory */home/user/wind*, with names *runa.cfl.1000*, *runa.cfl.2000*, etc.



**STAGES** — Multi-stage time stepping

|                                                                                                |
|------------------------------------------------------------------------------------------------|
| <b>STAGES</b> <i>ityp</i> [ <b>COEFFICIENTS</b> <i>x1 x2 ... xn</i> ] [ <i>zone_selector</i> ] |
|------------------------------------------------------------------------------------------------|

This keyword allows the user to control the Runge-Kutta time stepping option. The parameter *ityp* specifies the number of stages, and *x1 x2 ... xn* specify the coefficients. The following *ityp* values are available:

| <u><i>ityp</i></u> | <u>Stages</u>                             |
|--------------------|-------------------------------------------|
| 0                  | One stage, Euler implicit or explicit     |
| 1                  | Four stage, Jameson type                  |
| 5                  | Three stage, minimal storage, third order |

The default coefficients are:

| <u><i>ityp</i></u> | <u># Req'd</u> | <u>Defaults</u>         |
|--------------------|----------------|-------------------------|
| 0                  | —              | —                       |
| 1                  | 4              | 1/4, 1/3, 1/2, 1        |
| 5                  | 5              | 1/4, 8/15, 0, 5/12, 3/4 |

**TEST** — Non-production test options

`TEST number [mode]`

For purposes of program testing, the user may select one or more test options for any job. These options are selected by the keyword **TEST** (beginning in column 1) followed by one or two integer variables. The keyword and the integer variables must be separated by one or more blanks. The first integer selects the test option. The second integer, which is not mandatory, selects a mode for that option. If the second integer is omitted, a default mode is activated. Any number of valid test options may be selected. However, only one mode may be activated for each option. If an option is selected more than once, the mode identified in the last selection will be used.

The list of valid options and modes is provided in [Section 11](#).

**TSL | THIN SHEAR LAYER** — Thin shear layer option

{TSL | THIN} [SHEAR] [LAYER] *word1 word2 word3* [*nzn*]

This keyword turns off the viscous terms in specified coordinate directions. The parameters *word1*, *word2*, and *word3* are keywords controlling the viscous terms in the  $\xi$ -,  $\eta$ -, and  $\zeta$ -directions, respectively. They may have the following values:

| <u>Keyword</u> | <u>Meaning</u>                              |
|----------------|---------------------------------------------|
| VISCOUS        | Retain all viscous terms in this direction  |
| INVISCID       | Neglect all viscous terms in this direction |

*nzn* is the zone number for this specification. Omit *nzn* or set it to 0 to apply the current specification to all un-specified zones.

By default, when inviscid is set in a direction, the scalar implicit operator will be used in that direction (the full block implicit operator is used in viscous directions). This may be overwritten using the [IMPLICIT](#) keyword.

**TTSPEC** — Wall temperature and transition (block)

```

TTSPEC
  TYPE TEMPERATURE | TRANSITION
  MODE FILE filename | CURRENT
  ZONE zone_number
  SURFACE surfname
  SUBSET ALL
ENDTTSPEC

```

This keyword block is used when a point-by-point surface temperature boundary condition is being used, and/or when point-by-point boundary layer transition information is being specified. Surface temperature and transition data may be specified for any boundary surface in any zone.<sup>27</sup> The maximum total number of temperature/transition surfaces, combined, is determined by the Fortran parameter `MAXTTS`, currently equal to 20.

The normal procedure is to do an initial run, perhaps with just a few cycles, without specifying temperature/transition information, to create the initial common flow (*.cfl*) file. The temperature/transition data is then created and added to the *.cfl* file using the *tmptrn* utility.

For backward compatibility with WIND 2.0, however, the temperature/transition data may be read from separate ASCII files, one for each applicable surface, created with the older *tmptrn* utility. This would only be necessary during the initial (i.e., non-restart) run, since WIND will automatically store the data in the *.cfl* file, where it can be read during subsequent runs.

The various elements of the TTSPEC keyword block are defined as follows:

```
TYPE TEMPERATURE | TRANSITION
```

This keyword indicates whether a surface temperature distribution or boundary layer transition information is being specified.

```
MODE FILE filename | CURRENT
```

This keyword specifies where the surface temperature or transition data is stored.

**FILE *filename*** The data is stored in the ASCII file *filename*, created using the older *tmptrn* utility distributed with WIND 2.0. If *filename* is specified without an extension, an extension of *.dat* is assumed.

This option is included for backward compatibility only, using existing files created with the older *tmptrn* utility distributed with WIND 2.0, and is only needed for an initial (i.e., non-restart) run. Subsequent (i.e., restart) runs should use **MODE CURRENT**. New applications should use the latest *tmptrn* utility to write the temperature/transition data into the *.cfl* file directly.

**CURRENT** The data is stored in the flow (*.cfl*) file.

<sup>27</sup>Note, though, that although the WIND code allows temperature/transition data to be specified on any boundary surface, the *tmptrn* utility used to add the data to the *.cfl* file currently does not allow specification on the  $i_1$  or  $i_{max}$  surface.

ZONE *zone\_number*

This keyword specifies the number of the zone containing the boundary surface along which temperature or transition data is being specified.

SURFACE *surfname*

This keyword specifies the boundary surface in zone *zone\_number* along which temperature or transition data is being specified. Currently, the applicable surface names are I1, IMAX, J1, JMAX, K1 and KMAX, corresponding to the  $i_1$ ,  $i_{max}$ ,  $j_1$ ,  $j_{max}$ ,  $k_1$  and  $k_{max}$  surfaces, respectively.

SUBSET ALL

In the future, this keyword will allow temperature/transition information to be specified over a subset of the surface. Currently, SUBSET ALL is only used to update the counter of the number of temperature/transition surfaces, and must be specified once for each surface.

### Example

As an example, suppose we are specifying the wall temperature distribution along the  $j_1$  surface in zone 1, and the  $j_1$  and  $j_{max}$  surfaces in zone 2. In addition, suppose we're specifying boundary layer transition along the  $j_1$  surface in zone 1. If existing temperature/transition files are being used, created with the older *tmptrn* utility, the TTSPEC keyword block for the initial run might be

```

TTSPEC
  TYPE TEMPERATURE
    ZONE 1
      MODE FILE temp.z1j1.asc
      SURFACE J1
      SUBSET ALL
    ZONE 2
      MODE FILE temp.z2j1.asc
      SURFACE J1
      SUBSET ALL
      MODE FILE temp.z2jm.asc
      SURFACE JMAX
      SUBSET ALL
  TYPE TRANSITION
    ZONE 1
      MODE FILE tran.z1j1.asc
      SURFACE J1
      SUBSET ALL
ENDTTSPEC

```

Since the temperature/transition data will be written into the *.cfl* file during the initial run, later (i.e., restart) runs could use

```

TTSPEC
  TYPE TEMPERATURE
    ZONE 1
      MODE CURRENT
      SURFACE J1

```

```
        SUBSET ALL
    ZONE 2
        MODE CURRENT
        SURFACE J1
        SUBSET ALL
        MODE CURRENT
        SURFACE JMAX
        SUBSET ALL
TYPE TRANSITION
    ZONE 1
        MODE CURRENT
        SURFACE J1
        SUBSET ALL
ENDTTSPEC
```

*See Also:* [WALL TEMPERATURE](#)

**TURBULENCE** — Turbulence model selection

|                                                                                                                                   |
|-----------------------------------------------------------------------------------------------------------------------------------|
| <b>TURBULENCE</b> [MODEL] <i>model</i> [RC   ROTATION] [F2FIX <i>dmax</i> ] [ITERATIONS <i>iter</i> ]<br>[ <i>zone_selector</i> ] |
|-----------------------------------------------------------------------------------------------------------------------------------|

This keyword is used to request an inviscid or viscous solution, and to select a turbulence model for one or more zones. The parameter *model* must be one of the following keywords:

|                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| INVISCID   EULER   | Euler solution (scalar implicit)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| LAMINAR            | Viscous, laminar                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| CEBECI [SMITH]     | Viscous, Cebeci-Smith algebraic model                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| [BALDWIN] LOMAX    | Viscous, Baldwin-Lomax algebraic model                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| PDT                | Viscous, combination Baldwin-Lomax and P. D. Thomas algebraic shear layer model                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| [BALDWIN] BARTH    | Viscous, Baldwin-Barth one-equation model                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| SPALART [ALLMARAS] | Viscous, Spalart-Allmaras one-equation model. Additional keywords may be used with the Spalart-Allmaras model to specify the <a href="#">freestream turbulence level</a> (p. 168), and to specify use of the <a href="#">Spalart Detached Eddy Simulation (DES)</a> turbulence model (p. 169).                                                                                                                                                                                                                                                                                                                         |
| SST                | Viscous, two-equation Shear Stress Transport model. Additional keywords specific to the SST model are available, described starting on p. 169, for <a href="#">specifying freestream values of <math>k</math> and <math>\omega</math></a> . Another additional keyword may be used to specify use of a combined <a href="#">SST and Large Eddy Simulation (LES)</a> turbulence model, as described starting on p. 170. When the SST model is used for multi-zone problems, it must be used in all the zones. Using the regular SST model in some zones, and the combined SST and LES model in other zones, is allowed. |
| CHIEN [K-E]        | Viscous, Chien low-Reynolds-number two-equation model. <a href="#">Additional keywords specific to the Chien model</a> are also available, described starting on p. 171. When the Chien model is used for multi-zone problems, it must be used in all the zones.                                                                                                                                                                                                                                                                                                                                                       |

Note that, with the exceptions of the SST and Chien models, different turbulence models, or inviscid or laminar flow, may be specified in different zones. *However, you must specify a “default” turbulence model (or inviscid or laminar flow) in the input data file. WIND will stop if you do not.* By “default”, we mean without specifying zones.

For example, for a three-zone problem with inviscid flow in zone 1 and the Spalart-Allmaras turbulence model in zones 2 and 3, the following will *not* work, and the code will stop:

```
TURBULENCE INVISCID ZONE 1
TURBULENCE SPALART ZONE 2,3
```

Instead, one could specify the following, which will work:

```
TURBULENCE INVISCID
TURBULENCE SPALART ZONE 2,3
```

Additional parameters that may be used with the **TURBULENCE** keyword are:

|                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>RC   ROTATION</b>          | May be used with the Spalart-Allmaras one-equation model to include a correction for system rotation and streamline curvature                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>F2FIX <i>dmax</i></b>      | May be used with the SST two-equation model to specify a maximum distance from the wall, <i>dmax</i> , within which the $F_2$ term may be non-zero. Beyond that distance, the $F_2$ term is set to zero. $F_2$ is a blending function, designed to limit the effects of the shear stress transport term to regions near walls.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>ITERATIONS <i>iter</i></b> | <p>The WIND code organizes the equations to be solved into logical “groups” that are solved together. It also allows multiple iterations of a specific group (i.e., sub-iterations) for each “iteration per cycle”. (The number of iterations per cycle is set using the <a href="#">ITERATIONS</a> keyword.)</p> <p>For one- and two-equation turbulence models, this parameter allows the user to request <i>iter</i> sub-iterations of the turbulence model equation group for each “iteration per cycle”. If <a href="#">NAVIER-STOKES ITERATIONS</a> is defaulted, this corresponds to <i>iter</i> iterations of the turbulence model equations for each iteration of the mean flow equations.</p> <p>The default for <i>iter</i> is one, indicating that each “iteration per cycle” corresponds to one iteration of the turbulence model equations.</p> |

Note that for the one- and two-equation turbulence models, the turbulence model equations may be solved without simultaneously solving the Navier-Stokes equations. Of course, the turbulence variables depend on the mean flow field, so a reasonable mean flow solution must already exist.

As an example, suppose a mean flow solution has been computed using the SST turbulence model. The Chien  $k$ - $\epsilon$  variables consistent with the existing mean flow field may be computed by restarting from the SST solution, and solving just the Chien  $k$ - $\epsilon$  equations.

```
ITERATIONS PER CYCLE 2 ZONE ALL
NAVIER-STOKES ITERATIONS 0 ZONE ALL
TURBULENCE MODEL CHIEN ITER 5 ZONE ALL
```

The above keywords specify that, for all zones, there will be two iterations per cycle, with no Navier-Stokes sub-iterations and five Chien model sub-iterations for each “iteration per cycle”. There will thus be a total of ten iterations of the Chien  $k$ - $\epsilon$  equations in each zone prior to completing a cycle and exchanging information between zones.

See Also: [ITERATIONS](#), [NAVIER-STOKES ITERATIONS](#), [TEST 21](#)

## Spalart-Allmaras Model

```
FREE_ANUT anutinf
```

The separate keyword **FREE\_ANUT** may be used with the Spalart-Allmaras model to specify the freestream value of the eddy viscosity  $\nu_t$ . The default value is 5.0.

See Also: [TEST 21](#)



## Spalart DES Model

DES [CDES *cdes*] [*zone\_selector*]

The separate keyword **DES** may be used with the Spalart-Allmaras model to specify use of the Spalart Detached Eddy Simulation (DES) turbulence model. It is intended to improve the results for unsteady and massively separated flows. The DES model reduces to the standard Spalart-Allmaras model near viscous walls, where the grid is refined and has a large aspect ratio, but acts like a Large Eddy Simulation (LES) model away from the boundary, where the grid is coarser and has an aspect ratio of order one.

The input parameter *cdes* specifies the value of the coefficient  $C_{des}$  in the model. The default value is 0.65. Increasing  $C_{des}$  increases the size of the region in which the DES model reduces to the standard Spalart-Allmaras model. Details may be found in papers by [Spalart, Jou, Strelets, and Allmaras \(1997\)](#) and by [Shur, Spalart, Strelets, and Travin \(1999\)](#).

The DES model may only be used for unsteady (i.e., the time step is a constant) three-dimensional flows. It is zonal, however, so you can use the DES model in time-accurate mode in one zone, while using some other model in steady-state mode in the other zones. For example, a three-zone problem could use the regular Spalart-Allmaras model with a specified CFL number in zones 1 and 2, and the DES model with a specified time step in zone 3, using the following keywords:

```
TURBULENCE SPALART
DES ZONE 3
CFL# 1.3
TIMESTEP SECONDS 5.0E-6 ZONE 3
```

This capability can accelerate the solution convergence tremendously, especially for large configurations (10 to 20 million grid points) that would be impossible to run in time-accurate mode throughout the flow field.

## SST Model

FREE\_K *valk*  
 FREE\_OM *valom*

The separate keywords **FREE\_K** and **FREE\_OM** may be used with the SST model to specify dimensional freestream values of  $k$  and  $\omega$ . The following options are possible:

*valk* > 0      The turbulent kinetic energy  $k$  and the specific dissipation rate  $\omega$  are specified directly, with

$$k = valk \text{ (ft}^2\text{/sec}^2\text{)}$$

$$\omega = valom \text{ (1/sec)}$$

The turbulent viscosity  $\nu_t$  is then equal to  $k/\omega$ .

*valk* < 0      The turbulent kinetic energy  $k$  is set equal to *valk* percent of the “reference” kinetic energy  $U_\infty^2/2$ , where  $U_\infty$  is the freestream velocity. Thus

$$k = 0.01 |valk| \frac{U_\infty^2}{2}$$

The turbulent viscosity  $\nu_t$  is automatically set equal to  $0.001\nu_l$ , where  $\nu_l$  is the laminar viscosity, and the specific dissipation rate is computed as  $\omega = k/\nu_t$ .

$val\_om < 0$  The specific dissipation rate  $\omega$  is set equal to  $val\_om$  percent of  $U_\infty/L_{ref}$ , where  $U_\infty$  is the freestream velocity, and  $L_{ref}$  is the reference length from the grid (.cfd) file. Thus

$$\omega = 0.01 |val\_om| \frac{U_\infty}{L_{ref}}$$

The turbulent viscosity  $\nu_t$  is set to the same percentage of the laminar viscosity.

$$\nu_t = 0.01 |val\_om| \nu_l$$

The turbulent kinetic energy is then computed as  $k = \omega\nu_t$ .

The default, and recommended, values are

$$\begin{aligned}\omega &= 10U_\infty/L \\ k &= \omega\nu_t\end{aligned}$$

where the freestream turbulent viscosity  $\nu_t$  is arbitrarily set to  $0.001\nu_l$ .

Note that

- If  $val\_k > 0$ , a positive value must be specified for  $val\_om$ .
- If  $val\_k \leq 0$ ,  $val\_om$  should not be specified.
- If  $val\_om < 0$ , any value specified for  $val\_k$  is ignored.

Inflow turbulence levels may be specified for the SST model in the **ARBITRARY INFLOW** keyword block. If this is done, the **TURBULENCE** keyword must come before the **ARBITRARY INFLOW** keyword block in the input data (.dat) file.

See Also: **ARBITRARY INFLOW**

## Combined SST and LES Model

LESB [CBLES *cb*] [*zone\_selector*]

The separate keyword **LESB** may be used with the SST model to specify use of a combined SST and Large Eddy Simulation (LES) turbulence model. The intent is to improve predictions of complex flows in a real-world engineering environment, by allowing the use of LES methods with grids typical of those used with traditional Reynolds Averaged Navier Stokes models.

The combined model reduces to the standard SST model in high mean shear regions (e.g., near viscous walls), where the grid is refined and has a large aspect ratio unsuitable for LES models. As the grid is traversed away from high mean shear regions, it typically becomes coarser and more isotropic, and the combined model smoothly transitions to an LES model.

The input parameter *cb* specifies the value of the coefficient  $C_B$  in the model. The default value is 10.0. Increasing  $C_B$  increases the size of the region in which the combined model reduces to the standard SST model.

The combined model may only be used for unsteady flows (i.e., the time step is a constant). It is zonal, however, so you can use the combined model in time-accurate mode in one zone, while using the standard SST model in steady-state mode in the other zones. For example, a three-zone problem could use the standard SST model with a specified CFL number in zones 1 and 2, and the combined model with a specified time step in zone 3, using the following keywords:

```
TURBULENCE SST
LESB ZONE 3
CFL# 1.3
TIMESTEP SECONDS 5.0E-6 ZONE 3
```

### Chien $k$ - $\epsilon$ Model<sup>28</sup>

Several keywords may be used with the Chien  $k$ - $\epsilon$  model to control the initialization procedure, enhance its stability, and improve its accuracy in adverse pressure gradients and at high Mach numbers. For convenience, all keyword phrases associated with the Chien  $k$ - $\epsilon$  model begin with K-E.

K-E INITIALIZE [FROM] {EXISTING | EQUILIBRIUM | FREESTREAM}

This keyword determines how the turbulent transport variables for the Chien  $k$ - $\epsilon$  model ( $k$ ,  $\epsilon$ ,  $\mu_t$ ) will be initialized.

The first method, given by the **EXISTING** parameter, performs an “intelligent” initialization, based on the type of turbulence model (if any) used in the previous run. This is the default.

The second method, given by the **EQUILIBRIUM** parameter, uses an assumption of turbulent equilibrium, namely that the production,  $\Pi$ , of turbulent kinetic energy equals the rate of dissipation, together with an existing turbulent viscosity profile to initialize the  $k$  and  $\epsilon$  variables.

$$\rho\epsilon = \Pi/Re$$

$$\rho k = \sqrt{\frac{\rho\epsilon\mu_t}{C_\mu f_\mu Re}}$$

In order to use this technique, the user must first run a few thousand iterations using another eddy viscosity turbulence model. Initializing from an existing turbulent viscosity profile rather than uniform values aids somewhat in convergence and improves the stability of the model by reducing the dramatic changes in turbulence values that occur during the first few iterations after initialization.

The third method, given by the **FREESTREAM** parameter, initializes the turbulence variables to uniform values within each zone.

$$\rho k = \rho k_\infty$$

$$\rho\epsilon = \rho\epsilon_\infty$$

$$\mu_t = (\mu_t)_\infty$$

where the local density is used and the freestream conditions  $k_\infty$  and  $(\mu_t)_\infty$  may be specified with the keywords **K-E FREE\_K** and **K-E FREE\_MUT**.

K-E FREE\_K *val1*  
K-E FREE\_MUT *val2*

<sup>28</sup>The material in this section was originally written by Dennis Yoder of NASA Glenn Research Center.

These keywords may be used to specify the freestream turbulence values to be used when initializing the turbulence variables to uniform values (i.e., with the **K-E INITIALIZE FREESTREAM** keyword). The interpretation of the input depends on the signs of *val1* and *val2*.

For the turbulence kinetic energy  $k_\infty$ ,

$$k_\infty = \begin{cases} \frac{3}{2} I^2 u_\infty^2 & \text{for } val1 < 0, \text{ and where } |val1| = I \\ \bar{k}_\infty / \bar{a}_\infty^2 & \text{for } val1 > 0, \text{ and where } val1 = \bar{k}_\infty \text{ (ft/sec)}^2 \end{cases}$$

and for the turbulent viscosity  $(\mu_t)_\infty$ ,

$$(\mu_t)_\infty = \begin{cases} (\mu_t)_\infty & \text{for } val2 < 0, \text{ and where } |val2| = (\mu_t)_\infty \\ (\bar{\mu}_t)_\infty / \bar{\mu}_\infty & \text{for } val2 > 0, \text{ and where } val2 = (\bar{\mu}_t)_\infty \text{ (slug/ft-sec)} \end{cases}$$

where the overbar denotes a dimensional value.

If the K-E **FREE\_K** keyword is not used, or if *val1* = 0, then a default value of *val1* = -0.01 is used, which corresponds to a turbulence intensity *I* of 1%. If the K-E **FREE\_MUT** keyword is not used, or if *val2* = 0, then a default value of *val2* = -0.001 is used. This sets the freestream turbulent viscosity to be 0.001 times the freestream laminar viscosity. Note that for values greater than zero, the expected units for *val1* and *val2* are (ft/s)<sup>2</sup> and slug/ft-s, respectively.

These freestream values are only used when the *k-ε* model is initialized from uniform conditions. Currently, the model extrapolates to get values at all inflow and outflow boundaries. Thus, K-E **FREE\_K** and K-E **FREE\_MUT** can not be used to specify inflow conditions.

#### K-E REINITIALIZE

Under certain conditions, the *k-ε* information in the flow (.cfl) file can become “contaminated” and the model must be forced to reinitialize. Consider the following example. An attempt to initialize the *k-ε* model too soon results in poor *k* and *ε* profiles. So, the user decides to continue using another turbulence model until the flow is more well defined. When he later tries to reinitialize the *k-ε* model, he finds he has the same problem. This occurs because the .cfl file does not remove outdated information. Thus, when he specifies the *k-ε* model, the code reads the old *k* and *ε* information from the previous run.

Using the keyword phrase K-E **REINITIALIZE** signals the code to ignore the old *k-ε* information in the .cfl file and perform a fresh initialization. This command must be removed on subsequent runs or else the model will reinitialize itself each time.

#### K-E [TVD] ORDER {1|2|3}

This keyword sets the spatial order of accuracy of the TVD upwinding used in solving the *k-ε* equations. The default is first order.

#### K-E RELAX [FOR] val [ITERATIONS]

Updated values of *k*, *ε*, and  $\mu_t$  will be relaxed for *val* iterations (the default is 500) following the initialization. Relaxation of each of these variables reduces the amount they may change during any

single iteration. Immediately after initialization, the allowed changes are significantly reduced. This restriction is then gradually lifted as the last relaxation iteration is approached.

K-E [MAXIMUM] [TURBULENT] VISCOSITY *val1*  
 K-E [TURBULENT] [REFERENCE] VELOCITY *val2*

The  $k$ - $\epsilon$  model uses limiters within the interior of each zone to increase convergence and stability by capping the values of the turbulence quantities at both the high and low extremes. This is usually only necessary during the first few iterations after initialization, when the fluctuations in  $k$  and  $\epsilon$  tend to be the most dramatic.

Nondimensional values of the minimum limiters have been preset to small numbers. The minimum turbulent viscosity is set to 0.001,  $k_{min}$  is set to  $10^{-9}$ , and  $\epsilon_{min}$  is computed from the turbulent viscosity relation using an assumed reference density of 1, and  $f_\mu = 1$ .

The above keywords determine the maximum limiting values for the turbulence quantities. The first keyword sets the maximum turbulent viscosity to be *val1* (the default is 10,000) times the freestream viscosity. The second keyword sets the turbulent reference velocity equal to *val2* (the default is 1.0) times the freestream speed of sound. The maximum turbulent kinetic energy allowed is 10% of the kinetic energy of the turbulent reference velocity:

$$k_{max} = 0.10 \frac{u_{ref(k-\epsilon)}^2}{2}$$

The maximum dissipation rate is again computed from the turbulent viscosity relation.

The use of these limiters can be summarized as follows:

- If either  $k$  or  $\epsilon$  fall below the preset minimum values then both are reset to these values. This typically occurs in the freestream.
- If the turbulent kinetic energy exceeds  $k_{max}$ , then it is capped at this value. The dissipation rate is taken to be the larger of the current dissipation rate or  $\epsilon_{max}$ .
- If the turbulent viscosity exceeds  $(\mu_t)_{max}$ , then it is capped at this value and the turbulent kinetic energy is recomputed from the turbulent viscosity relation. The turbulent dissipation rate is left unchanged.

If the maximum limiters cause the turbulence variables to be capped within the flowfield, a warning message will be written to the list output (*.lis*) file. By using **CFPOST** to examine the normalized variable `mut muinf`, one can observe where these limiters are being used and adjust them using the above keywords. It is important that the turbulence values not be limited upon convergence.

K-E [VARIABLE] CMU {ON|OFF}

It is well known that the baseline  $k$ - $\epsilon$  model is poorly suited to adverse pressure gradient flows, such as those found in diffusers. Rodi and Scheuerer (1986) demonstrated that for these types of flows, the rate of dissipation near solid boundaries is too small relative to the rate of production of turbulent kinetic energy. This causes the model to overpredict skin friction and predict flows to be attached when experimental results show them to be separated.

The variable  $C_\mu$  formulation, which is derived from algebraic stress modeling, is designed to help remedy this problem by reducing the turbulent viscosity in regions of the flowfield where the production of turbulent kinetic energy is significantly larger than the rate of dissipation. The specific formulation used is:

$$C_\mu = \min \left( 0.09, \frac{0.10738(0.64286 + 0.19607R)}{[1 + 0.357(R - 1)]^2} \right)$$

As the ratio  $R$  of production to dissipation increases above 1, the coefficient  $C_\mu$  is reduced from its normal value of 0.09 to limit the turbulent viscosity.

The variable  $C_\mu$  option also provides added stability to the  $k$ - $\epsilon$  model, such as in the case of an airfoil, where the sudden deceleration of the flow near the leading edge would otherwise result in a significant rate of production. In regions of the flow where the turbulence is in equilibrium, i.e., where the production and dissipation are balanced, the turbulent viscosity remains unchanged.

The above keyword may be used to turn this option on or off (the default is ON).

K-E COMPRESSIBILITY [CORRECTION] {NONE | OFF | SARKAR | WILCOX | USER  $\alpha_k$   $M_{t_0}$ }

This keyword may be used to specify a compressibility correction, designed to enhance predictions at higher Mach numbers. The equation for the compressibility correction is

$$F(M_t) = \alpha_k \max(M_t^2 - M_{t_0}^2, 0)$$

The turbulent Mach number  $M_t$  is defined from  $M_t^2 = 2k/a^2$ , where  $a$  is the local speed of sound.

The parameters  $\alpha_k$  and  $M_{t_0}$  for the various options are defined in the following table.

| <u>Correction Type</u> | <u><math>\alpha_k</math></u> | <u><math>M_{t_0}</math></u> |
|------------------------|------------------------------|-----------------------------|
| NONE   OFF             | 0.0                          | 0.00                        |
| SARKAR                 | 1.0                          | 0.00                        |
| WILCOX                 | 1.5                          | 0.25                        |

Note that both NONE and OFF disable the compressibility correction, and that values of  $\alpha_k$  and  $M_{t_0}$  should only be included when using the USER option. The default is SARKAR.

Both the Sarkar and Wilcox compressibility corrections are designed to improve the prediction of compressible jet flows by including the compressible portion of the dissipation rate in the transport equation for the turbulent kinetic energy. These corrections use simple algebraic relations between the solenoidal and compressible dissipation rates. The effect of these corrections is to reduce the turbulent kinetic energy in regions where the flow is supersonic. In terms of supersonic jet predictions, this results in slower spreading rates, reduced mixing, and a longer core length.

See Also: [TEST 29](#), [TEST 51](#)

## Using CFPOST

Care must be taken when using the **CFPOST post-processing program** to examine turbulence variables. Since the common flow (.cfl) file does not remove outdated information, it may contain turbulence variables for several different turbulence models.

For example, suppose an initial case was run using the SST model, then restarted using the  $k$ - $\epsilon$  model. The .cfl file will contain values for the turbulent quantities  $k$ ,  $\omega$ ,  $\rho k$ ,  $\rho \epsilon$ , and  $\mu_t$ . To examine the  $k$ - $\epsilon$  variables ( $\rho k$ ,  $\rho \epsilon$ ), with the CFPOST **variables** command the user should specify

the variables `rho*k` and `rho*epsi`. Specifying just `k` in CFPOST will give the (older) turbulent kinetic energy from the SST model, not from the  $k$ - $\epsilon$  model.

The freestream parameters `kinf` and `einf` in the `.cfl` file will be those for whichever model was run most recently. They should therefore not be used to normalize turbulence quantities from earlier runs that used a different turbulence model.

In addition, since all the models use a turbulent viscosity, they share the variable  $\mu_t$ , which is specified in CFPOST as `mut`. The values displayed for  $\mu_t$  will be for whichever eddy viscosity model (even the Baldwin-Lomax model) was used most recently.

**TVD** — Total Variation Diminishing operator flag

TVD [FACTOR *factor* | OFF] [*zone\_selector*]

This keyword controls the TVD flux limiter in the explicit operator, which is helpful in preventing overshoots in flowfield properties in regions of high gradients. TVD is *enabled* by default.

**FACTOR** *factor* Specifies the “compression” parameter for the TVD algorithm, which controls the amount of limiting applied. The default compression parameter is the maximum value which assures TVD properties for the explicit operator being used (as specified by the [RHS](#) keyword), and are shown in the following table.

| <u>Explicit Operator</u> | <u><i>factor</i></u> |
|--------------------------|----------------------|
| First-order upwind       | Not applicable       |
| Second-order upwind      | 2                    |
| Second-order mixed       | 3                    |
| Third-order upwind       | 4                    |
| Second-order physical    | 3                    |
| Fourth- and fifth-order  | Not available        |

**OFF** Disables TVD for the specified zone(s)

*See Also:* [BOUNDARY TVD](#)



**VISCOSITY** — Specification of viscosity law

|                                                              |
|--------------------------------------------------------------|
| VISCOSITY {SUTHERLAND   WILKE   KEYE   CONSTANT <i>vis</i> } |
|--------------------------------------------------------------|

This keyword allows you to specify the method of computing the transport properties.

|                     |                                                                                            |
|---------------------|--------------------------------------------------------------------------------------------|
| SUTHERLAND          | Use Sutherland's law (ideal gases, $T > 180$ °R)                                           |
| WILKE               | Use Wilke's law for multi-species flow (real gases)                                        |
| KEYE                | Use a combination of Sutherland's law for $T \geq 180$ °R, and Keye's law for $T < 180$ °R |
| CONSTANT <i>vis</i> | Use a constant molecular viscosity of <i>vis</i> (slug/ft-sec)                             |

**WALL FUNCTION** — Specify the use of wall functions

**WALL FUNCTION** {on|off} [*zone\_selector*]

For turbulent flows, this keyword may be used to invoke wall function boundary conditions on viscous walls, using the White-Christoph law of the wall. Wall function boundary conditions allow calculations to be performed with fewer grid points, and generally with higher [CFL numbers](#).

A wall function boundary must lie on one of the boundaries of the selected zone, and may be an overlapped or internal boundary. The first grid point adjacent to the boundary must be within the log layer (roughly  $15 < y^+ < 100$ ). The recommended  $y^+$  value at the first point off the wall is about 50. The boundary condition will revert to a no-slip condition when the first grid point off the wall falls below  $y^+ = 15$ , indicating that it is within the laminar sublayer.

Note that the wall function boundary condition is used to eliminate the grid points that would otherwise be required to resolve the laminar sublayer and the logarithmic layer. The grid beyond the first point off the wall should be the same as for a run without wall functions, in order to properly resolve the rest of the boundary layer. The *cfssubset* utility may be useful when modifying an existing grid for use with wall functions.

The implementation of wall functions in WIND involves only the modification of the wall flux, and does not reset the values of any flow quantities.

Viscous forces on wall function boundaries calculated using the [LOADS](#) keyword should only be used for determining convergence. Correct values should be calculated from the *.cfl* file using the *integrate force* command in [CFPOST](#).

Based on limited test cases, the use of wall functions is not recommended if accurate predictions of wall heat transfer are required, or for flows with strong shock waves.

**WALL SLIP** — Iterations until no slip

`WALL SLIP [ITERATIONS] val [zone_selector]`

To minimize transients at the start of a WIND calculation, the velocity at no-slip boundaries is actually reduced from its initial value to the no-slip condition over a number of iterations. The **WALL SLIP** keyword allows the user to specify the number of iterations, given by *val*, on a zonal basis. The default value is 50 iterations.

## WALL TEMPERATURE — Specify wall temperature

WALL TEMPERATURE [ADIABATIC | *value* [EXTRAPOLATE [SPECIES]]] [*zone\_selector*]

This keyword allows you to specify a wall temperature on a zonal basis.

**ADIABATIC**                      Use an adiabatic wall boundary condition for temperature. This is the default.

*value*                              Use a constant wall temperature equal to *value* °R

**EXTRAPOLATE [SPECIES]**      Use extrapolation (zero gradient) for chemical species at the wall (finite rate chemistry only)

*See Also:* [TTSPEC](#)

## 11 Test Options

Several user-controlled options have been provided as an aid to modifying the WIND code. These options all may be selected by using the keyword **TEST** in the input data file. The test options typically control program features which are under test, and have not been accepted for production use. Each test option is of the form

**TEST** *number mode*

If *mode* is not described for a given test option, the user should use *mode* = 1 to activate that option.

As test options are accepted, they are “hard wired” into the code and the test option described here becomes meaningless. Test options may also be rejected based on trial runs. In that case, the option code described here will also become meaningless. For this reason, the list of valid options is not consecutive. Selection of an invalid option will be accepted by the WIND code, but will have no effect.

The various test options and modes are described in [Table 2](#). For each option, the subroutines referencing that option are listed in parentheses.

**Table 2: Non-Production Test Options**

| <i>number</i> | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1             | <p>Memory size check. Stop the run after displaying the memory requirements but before memory is allocated.</p> <p><u><i>mode</i></u>   <u>Result</u></p> <p><math>n &lt; 0</math>   Retrieve dimensions from the actual grid (.cgd) file</p> <p><math>n &gt; 0</math>   Retrieve dimensions from a text file with the following format:</p> <pre> #zones idim1 jdim1 kdim1 nfrpts1 ... idimn jdimn kdimn nfrptsn </pre> <p>Each of the <b>nfrpts</b> values is 0 except for overlapping grids. This mode allows one to determine how much memory WIND will require before generating the grid. Note that the name of the text file should end in .cgd, like the actual grid file.</p> <p>If <math> n  &gt; 1</math> an estimated workload balance for <math>n</math> processors will also be displayed.<br/>(mpinit, mpterm, opngrd, pswprt, wind, zinit)</p> |

*Continued on next page*

**Table 2: Non-Production Test Options** (*Continued*)

| number | Description                                                                                                                                                                                      |
|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2      | Designed for parallel processing data transfer debugging                                                                                                                                         |
|        | <u>mode</u> <u>Result</u>                                                                                                                                                                        |
|        | 1    Not used                                                                                                                                                                                    |
|        | 2    Don't read zonal boundary data ( <code>evrwbd</code> )                                                                                                                                      |
|        | 4    Don't solve zone ( <code>evsolv</code> )                                                                                                                                                    |
|        | 8    Don't write zonal data ( <code>evwzon</code> )                                                                                                                                              |
|        | 16   Don't write zonal boundary data ( <code>evrwbd</code> )                                                                                                                                     |
|        | 32   Not used                                                                                                                                                                                    |
|        | 64   Not used                                                                                                                                                                                    |
|        | 128   Not used                                                                                                                                                                                   |
|        | 256   Don't update boundary conditions ( <code>lpschm</code> )                                                                                                                                   |
|        | Set <i>mode</i> equal to the sum of the desired actions.                                                                                                                                         |
| 3      | Parallel processing task tracing                                                                                                                                                                 |
|        | <u>mode</u> <u>Action</u>                                                                                                                                                                        |
|        | 1    Trace event reads/writes ( <code>rwev</code> )                                                                                                                                              |
|        | 2    Trace file I/O ( <code>rwnh</code> , <code>rwstat</code> , <code>rwd</code> , <code>rwi</code> , <code>rwr</code> )                                                                         |
|        | 4    Trace network traffic ( <code>psexit</code> , <code>psrwev</code> , <code>psrwgv</code> , <code>psrwnh</code> , <code>psrwd</code> ,<br><code>psrwi</code> , <code>psrwr</code> )           |
|        | 8    Trace task begin/end ( <code>psspwn</code> , <code>tskbeg</code> , <code>tskidl</code> )                                                                                                    |
|        | 16   Print task queue for debugging ( <code>psqprrt</code> )                                                                                                                                     |
|        | Set <i>mode</i> equal to the sum of the desired actions. I.e., setting <i>mode</i> = 5 will trace both event reads/writes and network traffic.                                                   |
| 4      | Reserved for internal Boeing use in routines for hybrid unstructured grid capability ( <code>opngrd</code> )                                                                                     |
| 5      | Compute the L2 norm of the residual for the Navier-Stokes equations in the same way as the "old" code ( <code>l2norm</code> )                                                                    |
| 6      | Write <code>.cfl</code> file compatible with the "old" code ( <code>asnsx</code> , <code>asvisc</code> , <code>mpini2</code> )                                                                   |
| 7      | Do not use high performance C I/O interface ( <code>openf</code> , <code>rwstat</code> )                                                                                                         |
| 8      | Use Version 2 common files ( <code>mpinit</code> , <code>openf</code> , <code>rwstat</code> )                                                                                                    |
| 10     | Print the time step information (i.e., minimum/maximum CFL and $\Delta t$ ) into the <code>.lis</code> file every <i>mode</i> cycles, instead of just on the first cycle. ( <code>zsolv</code> ) |
| 17     | Use "new" Baldwin-Barth turbulence model ( <code>bbarth</code> )                                                                                                                                 |
| 20     | Non-dimensionalize $k$ and $\omega$ in the SST model the "old" way ( <code>aijkrg</code> , <code>aikeps</code> ,<br><code>sst</code> , <code>sstprtinp</code> , <code>sstpstprt</code> )         |

*Continued on next page*

**Table 2: Non-Production Test Options** (*Continued*)

| number | Description                                                                                                                                                                                                                                                                                                                                                                  |
|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 21     | Spalart-Allmaras turbulence model. (See the <a href="#">TURBULENCE</a> keyword.) ( <code>redimsa</code> , <code>sabound</code> , <code>sapreinp</code> , <code>sinut</code> , <code>spalart</code> )                                                                                                                                                                         |
|        | <u>mode</u> <u>Result</u>                                                                                                                                                                                                                                                                                                                                                    |
|        | 0   Use original 1992 model, with an $f_{t2}$ term for laminar stabilization and a default freestream value for $\nu_t$ of 5.0.                                                                                                                                                                                                                                              |
|        | 1   Like mode 0, except with “corrections” to the production and destruction terms. This is equivalent to the default model for WIND beta versions 4.15 to 4.92.                                                                                                                                                                                                             |
|        | 2   Like mode 1, except without the $f_{t2}$ term, and with a default freestream value for $\nu_t$ of 0.1. In addition, the initial value of the dependent variable $\tilde{\nu}$ is set to the freestream $\nu_t$ . This is equivalent to the default model for WIND versions prior to WIND beta 4.15, and includes a slight error that makes the model overly dissipative. |
| 25     | In the Baldwin-Lomax model, use $y^+$ based on wall vorticity ( <code>blomax</code> )                                                                                                                                                                                                                                                                                        |
| 26     | Use local values in $y^+$ damping for the Baldwin-Lomax, Cebeci-Smith, Baldwin-Barth, and Chien $k$ - $\epsilon$ models ( <code>bbdamp</code> , <code>blomax</code> , <code>cebeci</code> , <code>kepy2</code> )                                                                                                                                                             |
| 29     | For the Cebeci-Smith model, use $y^+$ based on wall vorticity. ( <code>algtur</code> , <code>cebeci</code> )                                                                                                                                                                                                                                                                 |
|        | For the Chien $k$ - $\epsilon$ model, TEST 29 is a production limiter ( <code>kelhssch</code> , <code>keprod</code> , <code>kerhssch</code> )                                                                                                                                                                                                                                |
|        | <u>mode</u> <u>Result</u>                                                                                                                                                                                                                                                                                                                                                    |
|        | 0   Production limited to $20 \times$ dissipation                                                                                                                                                                                                                                                                                                                            |
|        | 1   Production computed from vorticity, but not limited                                                                                                                                                                                                                                                                                                                      |
|        | 2   Production not limited                                                                                                                                                                                                                                                                                                                                                   |
|        | 3   Production computed from vorticity, then limited                                                                                                                                                                                                                                                                                                                         |
| 30     | Irrotational boundary condition at freestream inflow boundaries ( <code>bcfree</code> )                                                                                                                                                                                                                                                                                      |
| 40     | Old switch for compressibility correction in $k$ - $\epsilon$ model; use <a href="#">K-E COMPRESSIBILITY CORRECTION</a> instead. ( <code>keppstin</code> )                                                                                                                                                                                                                   |
| 46     | In the <a href="#">SST turbulence model</a> , in <a href="#">blowing regions</a> and <a href="#">bleed regions</a> with a specified negative bleed flow rate, set $\mu_{turb} = 10\mu_{lam}$ along the wall. ( <code>sstbound</code> )                                                                                                                                       |
| 47     | For the algebraic turbulence models, smooth turbulent viscosity in each $i$ -plane using simple averaging. <i>mode</i> = number of smoothing passes. ( <code>smtvis</code> )                                                                                                                                                                                                 |
| 48     | For the algebraic turbulence models, smooth turbulent viscosity in three dimensions using simple averaging. <i>mode</i> = number of smoothing passes. ( <code>smtvis</code> )                                                                                                                                                                                                |

*Continued on next page*

**Table 2: Non-Production Test Options** (*Continued*)

| number | Description                                                                                                                                                                                                                                                                                                                                                                       |
|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 49     | Modified Runge-Kutta smoothing (see the <b>SMOOTHING</b> keyword) ( <b>dampi</b> , <b>dampj</b> , <b>dampk</b> , <b>bdload</b> )                                                                                                                                                                                                                                                  |
|        | <u>mode</u> <u>Result</u>                                                                                                                                                                                                                                                                                                                                                         |
|        | 1    TVD-like parameters                                                                                                                                                                                                                                                                                                                                                          |
|        | 2    No pressure switch on second-order dissipation                                                                                                                                                                                                                                                                                                                               |
|        | 3    No pressure switch and an LES type filtering of nonlinear terms                                                                                                                                                                                                                                                                                                              |
| 51     | Limit the turbulent viscosity $\mu_T$ such that the maximum value of $\mu_T/(\mu_L)_\infty = mode \times 1000$ . Suggested range is $50 < mode < 100$ . Do not use this option with the Chien $k-\epsilon$ model; use <b>K-E MAXIMUM TURBULENT VISCOSITY</b> instead. ( <b>keppstin</b> , <b>mutlim</b> )                                                                         |
| 52     | When using <b>BLOW PLENUM</b> , print a warning when the plenum total pressure is automatically raised because it was less than the local static pressure ( <b>bcbled</b> )                                                                                                                                                                                                       |
| 53     | Allows no-slip in inviscid flow at “viscous wall” boundaries ( <b>bcwall</b> )                                                                                                                                                                                                                                                                                                    |
| 54     | Reserved for use at Boeing                                                                                                                                                                                                                                                                                                                                                        |
| 55     | Reserved for use at Boeing                                                                                                                                                                                                                                                                                                                                                        |
| 56     | No energy addition to fluid due to MFD equations ( <b>emdef</b> )                                                                                                                                                                                                                                                                                                                 |
| 57     | Implicit terms on for Spalart-Allmaras model ( <b>spalart</b> )                                                                                                                                                                                                                                                                                                                   |
| 58     | Store the Lorentz force in the <i>.cfl</i> file instead of the electric field ( <b>emdef</b> )                                                                                                                                                                                                                                                                                    |
| 59     | Apply <b>SMOOTHING</b> keyword values to the Spalart-Allmaras model as well as the mean flow solver. (Currently deactivated) ( <b>spalart</b> )                                                                                                                                                                                                                                   |
| 61     | When $mode = 2$ , all boundary conditions are applied, whether or not they're consistent with the <b>IBLANK</b> values. This only affects corners, where there are usually multiple boundary conditions. So, if a wall boundary at $j = 1$ meets an outflow boundary at $i = i_{max}$ , if <b>TEST 61 2</b> is specified, both boundary conditions are applied. ( <b>tdbcgs</b> ) |
| 63     | Eliminate the “fat” boundary cells in any coordinate direction. $mode = 1, 2$ , or $4$ indicates the $i, j$ , and $k$ direction, respectively. Set $mode$ equal to the sum of the desired directions. I.e., setting $mode = 5$ will eliminate the “fat” boundary cells in the $i$ and $k$ directions. ( <b>tdarea</b> )                                                           |
| 64     | Remove <b>dt</b> from <b>dq</b> when computing residuals ( <b>l2norm</b> )                                                                                                                                                                                                                                                                                                        |
| 65     | In marching solutions, lower the CFL number for the last marching step ( <b>zsolv</b> )                                                                                                                                                                                                                                                                                           |
| 66     | Don't update $\beta$ in <b>gas3</b> , for <b>ireal</b> = 2 and <b>ispec</b> = 2. This test option is not recommended but will decrease run time. ( <b>gas3</b> )                                                                                                                                                                                                                  |
| 70     | Tolerance level for converging gas properties $P$ , $\rho$ , or $T$ in the <b>gas</b> routines. Tolerance level = $(0.1)^{mode}$ . ( <b>gas1</b> , <b>gas2</b> , <b>gas3</b> , <b>gas4</b> )                                                                                                                                                                                      |
| 75     | ?? ( <b>prpold</b> , <b>tdimafk</b> , <b>tdimfu</b> )                                                                                                                                                                                                                                                                                                                             |

*Continued on next page*



**Table 2: Non-Production Test Options** (*Continued*)

| number | Description                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 77     | Set bleed limits for a specific F/A-18 15% scale sting and distortion model, for use with the <b>BLEED AEDC</b> keyword ( <b>bcaedc</b> )                                                                                                                                                                                                                                                                                                    |
|        | <u>mode</u> <u>Result</u>                                                                                                                                                                                                                                                                                                                                                                                                                    |
|        | 0   F/A-18E                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|        | 1   F/A-18A                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 84     | Use “old” viscous metric calculation ( <b>dsolv</b> , <b>vismet</b> )                                                                                                                                                                                                                                                                                                                                                                        |
| 85     | Check for zero volumes when computing viscous metrics ( <b>dsolv</b> , <b>vismet</b> )                                                                                                                                                                                                                                                                                                                                                       |
|        | <u>mode</u> <u>Result</u>                                                                                                                                                                                                                                                                                                                                                                                                                    |
|        | 1   Check; if $\leq 0$ print message and continue                                                                                                                                                                                                                                                                                                                                                                                            |
|        | 2   Check; if $\leq 0$ print message and stop                                                                                                                                                                                                                                                                                                                                                                                                |
| 87     | Freezes supersonic inflow at initial conditions ( <b>bcfree</b> )                                                                                                                                                                                                                                                                                                                                                                            |
| 88     | Bypass negative $T$ check in <b>tdgas1</b> . This is needed for chemistry if SHF (heat of formation) varies widely since we only have an old SHF to use to estimate $T$ . ( <b>lpschm</b> , <b>tdgas1</b> )                                                                                                                                                                                                                                  |
| 89     | Use “old” species flux correction method ( <b>gas1</b> )                                                                                                                                                                                                                                                                                                                                                                                     |
| 90     | Chemistry stuff ( <b>chinv</b> )                                                                                                                                                                                                                                                                                                                                                                                                             |
|        | <u>mode</u> <u>Result</u>                                                                                                                                                                                                                                                                                                                                                                                                                    |
|        | 0   Analytic chemistry Jacobian ( <b>ns</b> = 5 only)                                                                                                                                                                                                                                                                                                                                                                                        |
|        | 1   Householder chemistry Jacobian ( <b>ns</b> > 5)                                                                                                                                                                                                                                                                                                                                                                                          |
|        | 2   Solves chemistry source term explicitly                                                                                                                                                                                                                                                                                                                                                                                                  |
| 91     | Gas constant ( <b>chprtinp</b> , <b>therm1</b> , <b>thermp</b> )                                                                                                                                                                                                                                                                                                                                                                             |
|        | <u>mode</u> <u>Result</u>                                                                                                                                                                                                                                                                                                                                                                                                                    |
|        | 1 $\beta = \gamma = \beta_\infty$                                                                                                                                                                                                                                                                                                                                                                                                            |
|        | 2 $\beta = \gamma = 1.4$                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 92     | Utilize operator splitting for the reacting chemistry source terms to increase the stability of the integration, allowing more efficient solution of the coupled system. A 4th-order Pade approximation is used to integrate the reaction source terms, with <i>mode</i> setting the number of subiterations. Setting <i>mode</i> = 0 indicates no operator splitting. ( <b>chimplicit</b> , <b>chrhss</b> , <b>prpold</b> , <b>rhssrc</b> ) |
| 93     | Turn on chemistry diffusion ( <b>chrhsv</b> , <b>rhsvfl</b> , <b>tdutv1</b> )                                                                                                                                                                                                                                                                                                                                                                |
|        | <u>mode</u> <u>Result</u>                                                                                                                                                                                                                                                                                                                                                                                                                    |
|        | 1   Diffusion                                                                                                                                                                                                                                                                                                                                                                                                                                |
|        | 2   Add diffusion gradient to conduction term                                                                                                                                                                                                                                                                                                                                                                                                |
| 94     | Turn off implicit chemistry terms ( <b>tdimpl</b> , <b>tdutaa</b> )                                                                                                                                                                                                                                                                                                                                                                          |
| 95     | Turn off chemistry source term ( <b>rates</b> , <b>rates1</b> , <b>rates2</b> , <b>rates3</b> , <b>rates4</b> , <b>ratesa</b> , <b>ratesad1</b> , <b>ratesb</b> , <b>ratesf</b> )                                                                                                                                                                                                                                                            |
| 96     | Apply chemistry source term over <i>mode</i> iterations for finite-rate non-equilibrium chemistry ( <b>chrhss</b> )                                                                                                                                                                                                                                                                                                                          |

*Continued on next page*

**Table 2: Non-Production Test Options** (*Continued*)

| number      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |             |                       |   |                                                                          |   |                                                                                                                                                                                                                                          |   |                                                                                                                                                                                                                                             |   |                                                                                        |   |                                                                        |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|-----------------------|---|--------------------------------------------------------------------------|---|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|----------------------------------------------------------------------------------------|---|------------------------------------------------------------------------|
| 97          | P. D. Thomas turbulence model scanning direction. By default, WIND starts at viscous walls and moves into the field. This test option forces the code to calculate turbulent parameters from <i>any</i> boundary, in addition to walls. ( <b>algtur</b> )<br><table> <tr> <th><u>mode</u></th><th><u>Result</u></th></tr> <tr> <td>0</td><td>use <math>j</math> lines</td></tr> <tr> <td>1</td><td>use <math>k</math> lines</td></tr> <tr> <td>2</td><td>use <math>j</math> and <math>k</math> lines</td></tr> </table>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | <u>mode</u> | <u>Result</u>         | 0 | use $j$ lines                                                            | 1 | use $k$ lines                                                                                                                                                                                                                            | 2 | use $j$ and $k$ lines                                                                                                                                                                                                                       |   |                                                                                        |   |                                                                        |
| <u>mode</u> | <u>Result</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |             |                       |   |                                                                          |   |                                                                                                                                                                                                                                          |   |                                                                                                                                                                                                                                             |   |                                                                                        |   |                                                                        |
| 0           | use $j$ lines                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |             |                       |   |                                                                          |   |                                                                                                                                                                                                                                          |   |                                                                                                                                                                                                                                             |   |                                                                                        |   |                                                                        |
| 1           | use $k$ lines                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |             |                       |   |                                                                          |   |                                                                                                                                                                                                                                          |   |                                                                                                                                                                                                                                             |   |                                                                                        |   |                                                                        |
| 2           | use $j$ and $k$ lines                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |             |                       |   |                                                                          |   |                                                                                                                                                                                                                                          |   |                                                                                                                                                                                                                                             |   |                                                                                        |   |                                                                        |
| 98          | <i>mode</i> is the relaxation factor for the species mass fraction from the Liu and Vinokur (real gas) model. By increasing <i>mode</i> , the mass fraction relaxes faster. Default <i>mode</i> = 5. <i>mode</i> = 1 corresponds to instantaneous. ( <b>gas1</b> , <b>gas2</b> , <b>gas3</b> , <b>gas4</b> )                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |             |                       |   |                                                                          |   |                                                                                                                                                                                                                                          |   |                                                                                                                                                                                                                                             |   |                                                                                        |   |                                                                        |
| 99          | Initialize finite rate chemistry with Liu and Vinokur curve fits ( <b>gas2</b> )<br><table> <tr> <th><u>mode</u></th><th><u>Result</u></th></tr> <tr> <td>0</td><td>Do not track the species (valid to 50K?)</td></tr> <tr> <td>1</td><td>Track the species (valid to 10K?)</td></tr> </table>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | <u>mode</u> | <u>Result</u>         | 0 | Do not track the species (valid to 50K?)                                 | 1 | Track the species (valid to 10K?)                                                                                                                                                                                                        |   |                                                                                                                                                                                                                                             |   |                                                                                        |   |                                                                        |
| <u>mode</u> | <u>Result</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |             |                       |   |                                                                          |   |                                                                                                                                                                                                                                          |   |                                                                                                                                                                                                                                             |   |                                                                                        |   |                                                                        |
| 0           | Do not track the species (valid to 50K?)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |             |                       |   |                                                                          |   |                                                                                                                                                                                                                                          |   |                                                                                                                                                                                                                                             |   |                                                                                        |   |                                                                        |
| 1           | Track the species (valid to 10K?)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |             |                       |   |                                                                          |   |                                                                                                                                                                                                                                          |   |                                                                                                                                                                                                                                             |   |                                                                                        |   |                                                                        |
| 100         | Characteristic time-stepping boundary condition ( <b>bcfree</b> )<br><table> <tr> <th><u>mode</u></th><th><u>Result</u></th></tr> <tr> <td>0</td><td>Second order, with limit of <math>\Delta Q \leq Q/2</math></td></tr> <tr> <td>1</td><td>1st-order, original characteristic treatment</td></tr> <tr> <td>2</td><td>2nd-order, original characteristic treatment</td></tr> <tr> <td>3</td><td>1st-order, Roe's average characteristic treatment</td></tr> <tr> <td>4</td><td>2nd-order, Roe's average characteristic treatment</td></tr> </table>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | <u>mode</u> | <u>Result</u>         | 0 | Second order, with limit of $\Delta Q \leq Q/2$                          | 1 | 1st-order, original characteristic treatment                                                                                                                                                                                             | 2 | 2nd-order, original characteristic treatment                                                                                                                                                                                                | 3 | 1st-order, Roe's average characteristic treatment                                      | 4 | 2nd-order, Roe's average characteristic treatment                      |
| <u>mode</u> | <u>Result</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |             |                       |   |                                                                          |   |                                                                                                                                                                                                                                          |   |                                                                                                                                                                                                                                             |   |                                                                                        |   |                                                                        |
| 0           | Second order, with limit of $\Delta Q \leq Q/2$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |             |                       |   |                                                                          |   |                                                                                                                                                                                                                                          |   |                                                                                                                                                                                                                                             |   |                                                                                        |   |                                                                        |
| 1           | 1st-order, original characteristic treatment                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |             |                       |   |                                                                          |   |                                                                                                                                                                                                                                          |   |                                                                                                                                                                                                                                             |   |                                                                                        |   |                                                                        |
| 2           | 2nd-order, original characteristic treatment                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |             |                       |   |                                                                          |   |                                                                                                                                                                                                                                          |   |                                                                                                                                                                                                                                             |   |                                                                                        |   |                                                                        |
| 3           | 1st-order, Roe's average characteristic treatment                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |             |                       |   |                                                                          |   |                                                                                                                                                                                                                                          |   |                                                                                                                                                                                                                                             |   |                                                                                        |   |                                                                        |
| 4           | 2nd-order, Roe's average characteristic treatment                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |             |                       |   |                                                                          |   |                                                                                                                                                                                                                                          |   |                                                                                                                                                                                                                                             |   |                                                                                        |   |                                                                        |
| 102         | Use time-averaged back pressure for mass flow boundary condition ( <b>pdsmfr</b> )                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |             |                       |   |                                                                          |   |                                                                                                                                                                                                                                          |   |                                                                                                                                                                                                                                             |   |                                                                                        |   |                                                                        |
| 104         | Turn off implicit viscous terms ( <b>tdutv1</b> )                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |             |                       |   |                                                                          |   |                                                                                                                                                                                                                                          |   |                                                                                                                                                                                                                                             |   |                                                                                        |   |                                                                        |
| 105         | Time step type ( <b>prtinp</b> , <b>tdtmst</b> )<br><table> <tr> <th><u>mode</u></th><th><u>Time Step Type</u></th></tr> <tr> <td>0</td><td><math>\Delta t = \text{CFL} / \max(\lambda_\xi, \lambda_\eta, \lambda_\zeta)</math></td></tr> <tr> <td>1</td><td>Flow angle scaling, <math>\Delta t = \text{CFL} \times (f_\xi \Delta \xi + f_\eta \Delta \eta + f_\zeta \Delta \zeta)</math>, where<br/> <math>f_\xi = \sqrt{1 + \tan \theta + \tan \psi}</math><br/> <math>f_\eta = f_\xi \tan \theta</math><br/> <math>f_\zeta = f_\xi \tan \psi</math> </td></tr> <tr> <td>2</td><td>Velocity scaling, <math>\Delta t = \text{CFL} \times \min(f_\xi \Delta \xi, f_\eta \Delta \eta, f_\zeta \Delta \zeta)</math>, where<br/> <math>f_\xi = u/u_\xi /  u_\xi + c </math><br/> <math>f_\eta = u/u_\eta /  u_\eta + c </math><br/> <math>f_\zeta = u/u_\zeta /  u_\zeta + c </math> </td></tr> <tr> <td>3</td><td><math>\Delta t = \text{CFL} \times \min(\Delta \xi, \Delta \eta, \Delta \zeta) / ( u  + c)</math></td></tr> <tr> <td>4</td><td><math>\Delta t = \text{CFL} / (\lambda_\xi + \lambda_\eta + \lambda_\zeta)</math></td></tr> </table> | <u>mode</u> | <u>Time Step Type</u> | 0 | $\Delta t = \text{CFL} / \max(\lambda_\xi, \lambda_\eta, \lambda_\zeta)$ | 1 | Flow angle scaling, $\Delta t = \text{CFL} \times (f_\xi \Delta \xi + f_\eta \Delta \eta + f_\zeta \Delta \zeta)$ , where<br>$f_\xi = \sqrt{1 + \tan \theta + \tan \psi}$<br>$f_\eta = f_\xi \tan \theta$<br>$f_\zeta = f_\xi \tan \psi$ | 2 | Velocity scaling, $\Delta t = \text{CFL} \times \min(f_\xi \Delta \xi, f_\eta \Delta \eta, f_\zeta \Delta \zeta)$ , where<br>$f_\xi = u/u_\xi /  u_\xi + c $<br>$f_\eta = u/u_\eta /  u_\eta + c $<br>$f_\zeta = u/u_\zeta /  u_\zeta + c $ | 3 | $\Delta t = \text{CFL} \times \min(\Delta \xi, \Delta \eta, \Delta \zeta) / ( u  + c)$ | 4 | $\Delta t = \text{CFL} / (\lambda_\xi + \lambda_\eta + \lambda_\zeta)$ |
| <u>mode</u> | <u>Time Step Type</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |             |                       |   |                                                                          |   |                                                                                                                                                                                                                                          |   |                                                                                                                                                                                                                                             |   |                                                                                        |   |                                                                        |
| 0           | $\Delta t = \text{CFL} / \max(\lambda_\xi, \lambda_\eta, \lambda_\zeta)$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |             |                       |   |                                                                          |   |                                                                                                                                                                                                                                          |   |                                                                                                                                                                                                                                             |   |                                                                                        |   |                                                                        |
| 1           | Flow angle scaling, $\Delta t = \text{CFL} \times (f_\xi \Delta \xi + f_\eta \Delta \eta + f_\zeta \Delta \zeta)$ , where<br>$f_\xi = \sqrt{1 + \tan \theta + \tan \psi}$<br>$f_\eta = f_\xi \tan \theta$<br>$f_\zeta = f_\xi \tan \psi$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |             |                       |   |                                                                          |   |                                                                                                                                                                                                                                          |   |                                                                                                                                                                                                                                             |   |                                                                                        |   |                                                                        |
| 2           | Velocity scaling, $\Delta t = \text{CFL} \times \min(f_\xi \Delta \xi, f_\eta \Delta \eta, f_\zeta \Delta \zeta)$ , where<br>$f_\xi = u/u_\xi /  u_\xi + c $<br>$f_\eta = u/u_\eta /  u_\eta + c $<br>$f_\zeta = u/u_\zeta /  u_\zeta + c $                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |             |                       |   |                                                                          |   |                                                                                                                                                                                                                                          |   |                                                                                                                                                                                                                                             |   |                                                                                        |   |                                                                        |
| 3           | $\Delta t = \text{CFL} \times \min(\Delta \xi, \Delta \eta, \Delta \zeta) / ( u  + c)$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |             |                       |   |                                                                          |   |                                                                                                                                                                                                                                          |   |                                                                                                                                                                                                                                             |   |                                                                                        |   |                                                                        |
| 4           | $\Delta t = \text{CFL} / (\lambda_\xi + \lambda_\eta + \lambda_\zeta)$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |             |                       |   |                                                                          |   |                                                                                                                                                                                                                                          |   |                                                                                                                                                                                                                                             |   |                                                                                        |   |                                                                        |

*Continued on next page*

**Table 2: Non-Production Test Options** (*Continued*)

| number      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                                                    |                                    |                           |                                |     |                                                    |             |               |                                         |                                                              |     |                                         |      |                                                   |
|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------|------------------------------------|---------------------------|--------------------------------|-----|----------------------------------------------------|-------------|---------------|-----------------------------------------|--------------------------------------------------------------|-----|-----------------------------------------|------|---------------------------------------------------|
| 106         | Compute the time step at the start of every cycle (even when Newton time stepping is being used), instead of at the start of every iteration. ( <b>axflow</b> , <b>lpgrp</b> , <b>lpmg</b> , <b>redim</b> )                                                                                                                                                                                                                                                                                                                                                                          |                                                    |                                    |                           |                                |     |                                                    |             |               |                                         |                                                              |     |                                         |      |                                                   |
| 108         | Extrapolate freestream outflow ( <b>bcfree</b> )<br><table><tr><th><u>mode</u></th><th><u>Mach</u></th><th><u>Outflow Conditions</u></th></tr><tr><td>0</td><td>&lt; 1</td><td>Hold upstream running characteristic at freestream</td></tr><tr><td></td><td>&gt; 1</td><td>Extrapolate all, even in boundary layer</td></tr><tr><td>1</td><td>All</td><td>Extrapolate all, even in boundary layer</td></tr></table>                                                                                                                                                                  | <u>mode</u>                                        | <u>Mach</u>                        | <u>Outflow Conditions</u> | 0                              | < 1 | Hold upstream running characteristic at freestream |             | > 1           | Extrapolate all, even in boundary layer | 1                                                            | All | Extrapolate all, even in boundary layer |      |                                                   |
| <u>mode</u> | <u>Mach</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | <u>Outflow Conditions</u>                          |                                    |                           |                                |     |                                                    |             |               |                                         |                                                              |     |                                         |      |                                                   |
| 0           | < 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Hold upstream running characteristic at freestream |                                    |                           |                                |     |                                                    |             |               |                                         |                                                              |     |                                         |      |                                                   |
|             | > 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Extrapolate all, even in boundary layer            |                                    |                           |                                |     |                                                    |             |               |                                         |                                                              |     |                                         |      |                                                   |
| 1           | All                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Extrapolate all, even in boundary layer            |                                    |                           |                                |     |                                                    |             |               |                                         |                                                              |     |                                         |      |                                                   |
| 109         | Boundary flux treatment ( <b>roewal</b> , <b>tdup1</b> )<br>For <b>tdup1</b> :<br><table><tr><th><u>mode</u></th><th><u>Result</u></th></tr><tr><td>0</td><td>Conservative</td></tr><tr><td>1</td><td>Upwind extrapolation from interior</td></tr></table><br>For <b>roewal</b> :<br><table><tr><th><u>mode</u></th><th><u>Result</u></th></tr><tr><td>0</td><td>Characteristic inflow, conservation if flow parallel to wall</td></tr><tr><td>1</td><td>Characteristic regardless</td></tr><tr><td>1000</td><td>Use conservative wall treatment at all boundaries</td></tr></table> | <u>mode</u>                                        | <u>Result</u>                      | 0                         | Conservative                   | 1   | Upwind extrapolation from interior                 | <u>mode</u> | <u>Result</u> | 0                                       | Characteristic inflow, conservation if flow parallel to wall | 1   | Characteristic regardless               | 1000 | Use conservative wall treatment at all boundaries |
| <u>mode</u> | <u>Result</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                                    |                                    |                           |                                |     |                                                    |             |               |                                         |                                                              |     |                                         |      |                                                   |
| 0           | Conservative                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                                                    |                                    |                           |                                |     |                                                    |             |               |                                         |                                                              |     |                                         |      |                                                   |
| 1           | Upwind extrapolation from interior                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                                    |                                    |                           |                                |     |                                                    |             |               |                                         |                                                              |     |                                         |      |                                                   |
| <u>mode</u> | <u>Result</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                                    |                                    |                           |                                |     |                                                    |             |               |                                         |                                                              |     |                                         |      |                                                   |
| 0           | Characteristic inflow, conservation if flow parallel to wall                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                                                    |                                    |                           |                                |     |                                                    |             |               |                                         |                                                              |     |                                         |      |                                                   |
| 1           | Characteristic regardless                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                    |                                    |                           |                                |     |                                                    |             |               |                                         |                                                              |     |                                         |      |                                                   |
| 1000        | Use conservative wall treatment at all boundaries                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                                                    |                                    |                           |                                |     |                                                    |             |               |                                         |                                                              |     |                                         |      |                                                   |
| 110         | Grid area variation limiting. Not allowed for <b>iorder</b> > 24, i.e., for the following Roe and Van Leer explicit operators: third-order fully upwind, fourth-order upwind-biased, fourth-order central, and fifth-order upwind-biased. ( <b>roecof</b> )<br><table><tr><th><u>mode</u></th><th><u>A<sub>2</sub>/A<sub>1</sub></u></th></tr><tr><td>0</td><td>∞</td></tr><tr><td>1</td><td>2.0</td></tr><tr><td>2</td><td>1.5</td></tr><tr><td>3</td><td>1.33</td></tr><tr><td>4</td><td>1.1</td></tr></table>                                                                     | <u>mode</u>                                        | <u>A<sub>2</sub>/A<sub>1</sub></u> | 0                         | ∞                              | 1   | 2.0                                                | 2           | 1.5           | 3                                       | 1.33                                                         | 4   | 1.1                                     |      |                                                   |
| <u>mode</u> | <u>A<sub>2</sub>/A<sub>1</sub></u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                                    |                                    |                           |                                |     |                                                    |             |               |                                         |                                                              |     |                                         |      |                                                   |
| 0           | ∞                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                                                    |                                    |                           |                                |     |                                                    |             |               |                                         |                                                              |     |                                         |      |                                                   |
| 1           | 2.0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                                                    |                                    |                           |                                |     |                                                    |             |               |                                         |                                                              |     |                                         |      |                                                   |
| 2           | 1.5                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                                                    |                                    |                           |                                |     |                                                    |             |               |                                         |                                                              |     |                                         |      |                                                   |
| 3           | 1.33                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                                                    |                                    |                           |                                |     |                                                    |             |               |                                         |                                                              |     |                                         |      |                                                   |
| 4           | 1.1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                                                    |                                    |                           |                                |     |                                                    |             |               |                                         |                                                              |     |                                         |      |                                                   |
| 111         | Singular matrix check ( <b>jacpr4</b> , <b>jacpr5</b> , <b>jacprg</b> , <b>tdsol4</b> , <b>tdsol9</b> , <b>tdsol11</b> , <b>tdsolg</b> , <b>tdsolv</b> )<br><table><tr><th><u>mode</u></th><th><u>Result</u></th></tr><tr><td>1</td><td>Check, but don't print results</td></tr><tr><td>2</td><td>Don't check</td></tr></table>                                                                                                                                                                                                                                                      | <u>mode</u>                                        | <u>Result</u>                      | 1                         | Check, but don't print results | 2   | Don't check                                        |             |               |                                         |                                                              |     |                                         |      |                                                   |
| <u>mode</u> | <u>Result</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                                    |                                    |                           |                                |     |                                                    |             |               |                                         |                                                              |     |                                         |      |                                                   |
| 1           | Check, but don't print results                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                                                    |                                    |                           |                                |     |                                                    |             |               |                                         |                                                              |     |                                         |      |                                                   |
| 2           | Don't check                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                                                    |                                    |                           |                                |     |                                                    |             |               |                                         |                                                              |     |                                         |      |                                                   |
| 112         | Corrected upwind scheme at boundaries. Defaults to corrected scheme, <i>mode</i> > 0 uses second order smoothing with <i>mode</i> /1000 as the smoothing level. Users should not use this option. ( <b>rhsupw</b> )                                                                                                                                                                                                                                                                                                                                                                  |                                                    |                                    |                           |                                |     |                                                    |             |               |                                         |                                                              |     |                                         |      |                                                   |

*Continued on next page*

**Table 2: Non-Production Test Options** (*Continued*)

| number | Description                                                                                                                                                                                                        |
|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 113    | Check for reverse flow at inflow and outflow boundaries ( <code>bcconf</code> , <code>bcfree</code> )                                                                                                              |
|        | <u>mode</u> <u>Result</u>                                                                                                                                                                                          |
|        | 0, 1   Print a warning message and continue                                                                                                                                                                        |
|        | 2   Print an error message and stop                                                                                                                                                                                |
| 114    | Central difference $\zeta$ operator ( <code>tdup1</code> )                                                                                                                                                         |
|        | <u>mode</u> <u>Result</u>                                                                                                                                                                                          |
|        | 0   Upwind                                                                                                                                                                                                         |
|        | $n$ Central smoothing coefficient = $n/1000$                                                                                                                                                                       |
| 115    | Do not rescale inviscid wall total velocity to equal adjacent value, just subtract the normal component from the adjacent value ( <code>bcwall</code> )                                                            |
| 116    | Set inward pointing normal to zero in <code>tdbcm1</code> at unknown grid topology points ( <code>tdbcm1</code> )                                                                                                  |
| 117    | Freeze inflow boundaries, even in subsonic flow ( <code>bcfree</code> )                                                                                                                                            |
|        | <u>mode</u> <u>Result</u>                                                                                                                                                                                          |
|        | 1   Freeze all inflow                                                                                                                                                                                              |
|        | 2   Freeze only arbitrary inflow points                                                                                                                                                                            |
|        | 3   Freeze characteristics on all $i = 1$ boundaries                                                                                                                                                               |
| 118    | Singular axis fix ( <code>bcpinw</code> , <code>bcsing</code> , <code>kerot</code> , <code>radavg</code> , <code>sngthrm</code> )                                                                                  |
|        | <u>mode</u> <u>Result</u>                                                                                                                                                                                          |
|        | 0   Average conservation variables                                                                                                                                                                                 |
|        | 1   Average primitive variables                                                                                                                                                                                    |
| 121    | Under-Relaxation of points adjacent to singular axis ( <code>bcpinw</code> , <code>bcsing</code> , <code>kebc</code> , <code>relnsg</code> , <code>sabound</code> , <code>sngthrm</code> , <code>sstbound</code> ) |
|        | <u>mode</u> <u>Result</u>                                                                                                                                                                                          |
|        | 0   Value on axis is a radius-weighted average of the values at the adjacent points; values at the adjacent points are unchanged                                                                                   |
|        | $n$ Value on axis computed as for mode 1; values at the adjacent points are computed from                                                                                                                          |
|        | $F_{adj} = (1 - r)F_{adj} + rF_{axis}$                                                                                                                                                                             |
|        | where $F_{adj}$ is the value at the adjacent point, $F_{axis}$ is the axis value, and $r = n/1000$ .                                                                                                               |
| 122    | Allow left-handed coordinates ( <code>tdarea</code> )                                                                                                                                                              |
| 123    | Track back pressure, mass flow, and integrated total pressure with outflow boundary conditions ( <code>bcconf</code> , <code>pdsfmfr</code> )                                                                      |
| 124    | Write convergence information to list output ( <code>.lis</code> ) file every iteration instead of every cycle ( <code>lpschm</code> )                                                                             |

*Continued on next page*

**Table 2: Non-Production Test Options** (*Continued*)

| number      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                          |             |        |   |                               |      |                                                             |   |                                                       |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|--------|---|-------------------------------|------|-------------------------------------------------------------|---|-------------------------------------------------------|
| 125         | Write time history output, specified using the <b>HISTORY</b> keyword, into the <i>.lis</i> file instead of a <i>.cth</i> file. This can significantly increase the size of the <i>.lis</i> file, but allows time history to be used in parallel mode. <i>resplt</i> can be used to create a GENPLOT file that may then be plotted using CFPOST. ( <b>maklut</b> , <b>tdsvth</b> , <b>tdwrth</b> , <b>thinit</b> , <b>thpstinp</b> , <b>thterm</b> ) |             |        |   |                               |      |                                                             |   |                                                       |
| 126         | Compressibility correction to Baldwin-Lomax turbulence model ( <b>blomax</b> )<br><table> <tr> <th><i>mode</i></th><th>Result</th></tr> <tr> <td>0</td><td>No compressibility correction</td></tr> <tr> <td>1</td><td><math>\kappa = 0.0180</math> for Baldwin-Lomax model (CFL3D uses this)</td></tr> </table>                                                                                                                                      | <i>mode</i> | Result | 0 | No compressibility correction | 1    | $\kappa = 0.0180$ for Baldwin-Lomax model (CFL3D uses this) |   |                                                       |
| <i>mode</i> | Result                                                                                                                                                                                                                                                                                                                                                                                                                                               |             |        |   |                               |      |                                                             |   |                                                       |
| 0           | No compressibility correction                                                                                                                                                                                                                                                                                                                                                                                                                        |             |        |   |                               |      |                                                             |   |                                                       |
| 1           | $\kappa = 0.0180$ for Baldwin-Lomax model (CFL3D uses this)                                                                                                                                                                                                                                                                                                                                                                                          |             |        |   |                               |      |                                                             |   |                                                       |
| 127         | Scale printed residual by maximum residual over all time steps ( <b>lpschm</b> )                                                                                                                                                                                                                                                                                                                                                                     |             |        |   |                               |      |                                                             |   |                                                       |
| 128         | Check the L2 norm of the residual for convergence instead of the maximum residual ( <b>l2norm</b> )                                                                                                                                                                                                                                                                                                                                                  |             |        |   |                               |      |                                                             |   |                                                       |
| 130         | Roe scheme physical space extrapolation ( <b>bcfree</b> )                                                                                                                                                                                                                                                                                                                                                                                            |             |        |   |                               |      |                                                             |   |                                                       |
| 131         | For boundary layers on $j = 1$ walls, set the time step in the boundary layer to a (larger) “outer” value, defined as the value at $j = mode$ . I.e., $(\Delta t)_j = (\Delta t)_{mode}$ for $j < mode$ . ( <b>tdtmst</b> )                                                                                                                                                                                                                          |             |        |   |                               |      |                                                             |   |                                                       |
| 132         | Renormalize, changing from total to static values. Normalizing values in the <i>.cfl</i> file are unchanged. ( <b>redim</b> )                                                                                                                                                                                                                                                                                                                        |             |        |   |                               |      |                                                             |   |                                                       |
| 133         | Print multi-grid sub-iteration convergence data (currently deactivated) ( <b>lpschm</b> )                                                                                                                                                                                                                                                                                                                                                            |             |        |   |                               |      |                                                             |   |                                                       |
| 134         | Second order characteristic extrapolation for adjacent conditions ( <b>bcfree</b> )<br><table> <tr> <th><i>mode</i></th><th>Result</th></tr> <tr> <td>2</td><td>First order</td></tr> <tr> <td>0, 1</td><td>Second order, with limit of <math>\Delta Q \leq Q/2</math></td></tr> <tr> <td>n</td><td>Second order, with limit of <math>\Delta Q \leq (n/1000)Q</math></td></tr> </table>                                                              | <i>mode</i> | Result | 2 | First order                   | 0, 1 | Second order, with limit of $\Delta Q \leq Q/2$             | n | Second order, with limit of $\Delta Q \leq (n/1000)Q$ |
| <i>mode</i> | Result                                                                                                                                                                                                                                                                                                                                                                                                                                               |             |        |   |                               |      |                                                             |   |                                                       |
| 2           | First order                                                                                                                                                                                                                                                                                                                                                                                                                                          |             |        |   |                               |      |                                                             |   |                                                       |
| 0, 1        | Second order, with limit of $\Delta Q \leq Q/2$                                                                                                                                                                                                                                                                                                                                                                                                      |             |        |   |                               |      |                                                             |   |                                                       |
| n           | Second order, with limit of $\Delta Q \leq (n/1000)Q$                                                                                                                                                                                                                                                                                                                                                                                                |             |        |   |                               |      |                                                             |   |                                                       |
| 135         | Resets the time step using a weighting function between the ordinary Euler CFL number and a new “viscous CFL number”, for convergence acceleration in viscous layers. The viscous CFL number is set to $mode/1000$ . Limited testing indicates that a value of $mode = 50$ is stable and increases the time step near the wall by at least an order of magnitude. ( <b>tdtmst</b> )                                                                  |             |        |   |                               |      |                                                             |   |                                                       |

*Continued on next page*

**Table 2: Non-Production Test Options** (*Continued*)

| number | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 136    | Divergence checker, $mode = n_1 + 10n_2$ , where ( <b>lpgrp</b> )<br><u><math>n_1</math></u> <u>Divergence Definition</u><br>1   Max residual > 1.0, L2 norm increasing<br>2   Max residual > 5.0, L2 norm increasing<br>3   Max residual > 10.0, L2 norm increasing<br>and<br><u><math>n_2</math></u> <u>Action Taken When Diverging</u><br>1   Terminate iteration for current cycle<br>2   Abort run<br>3   Reduce CFL number by 1/2<br>4   Reduce CFL number by 1/2 and terminate iteration for current cycle |
| 137    | Butt line interpolation region for <b>USERSPEC</b> ; smear <b>USERSPEC</b> conditions over $0.001 \times$ butt line at minimum and maximum butt line ( <b>uspeci</b> )<br><u><math>mode</math></u> <u>Result</u><br>0   No interpolation<br>n $n = 0.001 \times$ butt line for interpolation                                                                                                                                                                                                                      |
| 138    | Use large cell Jacobians at boundaries ( <b>bcwall</b> , <b>chrhsv</b> , <b>nsrhsv</b> , <b>tdarea</b> , <b>vismet</b> )<br><u><math>mode</math></u> <u>Result</u><br>$\leq 1$ Use large cells<br>2   Use large cells, central difference Jacobian<br>3   Throw out half cell at boundaries<br>5   Solve $\partial P / \partial n$ equation at walls                                                                                                                                                              |
| 139    | ?? ( <b>tdup1</b> )                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 140    | Use first-order differencing for computing $\partial(u, v, w) / \partial \xi$ term in vorticity used in turbulence models ( <b>vortcy</b> )                                                                                                                                                                                                                                                                                                                                                                       |
| 141    | 2nd-order wall boundary conditions (explicit) ( <b>bcwall</b> )<br><u><math>mode</math></u> <u>Result</u><br>1   Second order $\partial p / \partial n$ , $\partial T / \partial n$ , and $\partial u_{tan} / \partial n$<br>2   Second order $\partial p / \partial n$ and $\partial T / \partial n$ , but not $\partial u_{tan} / \partial n$                                                                                                                                                                   |
| 147    | ?? (currently deactivated) ( <b>turpdt</b> )                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 148    | ?? (currently deactivated) ( <b>turpdt</b> )                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |

*Continued on next page*

**Table 2: Non-Production Test Options** (*Continued*)

| number      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |             |               |   |                                                                                                       |   |                                                         |   |                  |   |                                         |   |             |   |             |   |             |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|---------------|---|-------------------------------------------------------------------------------------------------------|---|---------------------------------------------------------|---|------------------|---|-----------------------------------------|---|-------------|---|-------------|---|-------------|
| 150         | Singular axis on symmetry planes. When symmetry plane test fails, zero this component of velocity. ( <b>bcsing</b> )                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |             |               |   |                                                                                                       |   |                                                         |   |                  |   |                                         |   |             |   |             |   |             |
|             | <table> <tr> <th><u>mode</u></th><th><u>Result</u></th></tr> <tr> <td>1</td><td><math>u = 0</math></td></tr> <tr> <td>2</td><td><math>v = 0</math></td></tr> <tr> <td>3</td><td><math>w = 0</math></td></tr> <tr> <td>4</td><td>do not zero any component (use average)</td></tr> <tr> <td>5</td><td><math>v = w = 0</math></td></tr> <tr> <td>6</td><td><math>u = w = 0</math></td></tr> <tr> <td>7</td><td><math>u = v = 0</math></td></tr> </table>                                                                                                                                                                                                                                                                                                                                       | <u>mode</u> | <u>Result</u> | 1 | $u = 0$                                                                                               | 2 | $v = 0$                                                 | 3 | $w = 0$          | 4 | do not zero any component (use average) | 5 | $v = w = 0$ | 6 | $u = w = 0$ | 7 | $u = v = 0$ |
| <u>mode</u> | <u>Result</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |             |               |   |                                                                                                       |   |                                                         |   |                  |   |                                         |   |             |   |             |   |             |
| 1           | $u = 0$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |             |               |   |                                                                                                       |   |                                                         |   |                  |   |                                         |   |             |   |             |   |             |
| 2           | $v = 0$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |             |               |   |                                                                                                       |   |                                                         |   |                  |   |                                         |   |             |   |             |   |             |
| 3           | $w = 0$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |             |               |   |                                                                                                       |   |                                                         |   |                  |   |                                         |   |             |   |             |   |             |
| 4           | do not zero any component (use average)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |             |               |   |                                                                                                       |   |                                                         |   |                  |   |                                         |   |             |   |             |   |             |
| 5           | $v = w = 0$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |             |               |   |                                                                                                       |   |                                                         |   |                  |   |                                         |   |             |   |             |   |             |
| 6           | $u = w = 0$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |             |               |   |                                                                                                       |   |                                                         |   |                  |   |                                         |   |             |   |             |   |             |
| 7           | $u = v = 0$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |             |               |   |                                                                                                       |   |                                                         |   |                  |   |                                         |   |             |   |             |   |             |
| 151         | For a singular axis, the value on the axis is a radius-weighted average of the values two points away from the axis, instead of the values at the adjacent points; values at the adjacent points are set to the average of the axis value and the value two points away from the axis. E.g., for a singular axis at $j = 1$ , with the $k$ direction “circumferential”, the value on the axis is a radius-weighted average of the values at $j = 3$ , instead of at $j = 2$ . Then for each $k$ , the value at $j = 2$ is set to the average of the values at $j = 1$ and $j = 3$ . This test option overrides <a href="#">TEST 121</a> . ( <b>bcpinw</b> , <b>bcsing</b> , <b>kebc</b> , <b>kerot</b> , <b>radavg</b> , <b>relnsg</b> , <b>sabound</b> , <b>sngthrm</b> , <b>sstbound</b> ) |             |               |   |                                                                                                       |   |                                                         |   |                  |   |                                         |   |             |   |             |   |             |
| 153         | Iteration frequency for updating pressure at outflow boundaries. Default is 5. ( <b>bcconf</b> )                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |             |               |   |                                                                                                       |   |                                                         |   |                  |   |                                         |   |             |   |             |   |             |
| 157         | <b>USERSPEC</b> inflow ( <b>uspeci</b> )                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |             |               |   |                                                                                                       |   |                                                         |   |                  |   |                                         |   |             |   |             |   |             |
|             | <table> <tr> <th><u>mode</u></th><th><u>Result</u></th></tr> <tr> <td>1</td><td>Put <b>USERSPEC</b> inflow at all points in the buttline range. Do not check for above/below vehicle.</td></tr> <tr> <td>2</td><td>Same as mode 1, but also ignore fuselage station check.</td></tr> </table>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | <u>mode</u> | <u>Result</u> | 1 | Put <b>USERSPEC</b> inflow at all points in the buttline range. Do not check for above/below vehicle. | 2 | Same as mode 1, but also ignore fuselage station check. |   |                  |   |                                         |   |             |   |             |   |             |
| <u>mode</u> | <u>Result</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |             |               |   |                                                                                                       |   |                                                         |   |                  |   |                                         |   |             |   |             |   |             |
| 1           | Put <b>USERSPEC</b> inflow at all points in the buttline range. Do not check for above/below vehicle.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |             |               |   |                                                                                                       |   |                                                         |   |                  |   |                                         |   |             |   |             |   |             |
| 2           | Same as mode 1, but also ignore fuselage station check.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |             |               |   |                                                                                                       |   |                                                         |   |                  |   |                                         |   |             |   |             |   |             |
| 158         | Reserved for internal Boeing use in routines for hybrid unstructured grid capability ( <b>opngrd</b> , <b>pstinp</b> )                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |             |               |   |                                                                                                       |   |                                                         |   |                  |   |                                         |   |             |   |             |   |             |
| 160         | Pressure correction factor = $mode/1000$ for specified mass flow boundary ( <b>pdsmfr</b> )                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |             |               |   |                                                                                                       |   |                                                         |   |                  |   |                                         |   |             |   |             |   |             |
| 162         | Cebeci-Smith boundary layer edge definition ( <b>cebeci</b> )                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |             |               |   |                                                                                                       |   |                                                         |   |                  |   |                                         |   |             |   |             |   |             |
|             | <table> <tr> <th><u>mode</u></th><th><u>Result</u></th></tr> <tr> <td>0</td><td>1.0% change in <math>U_{total}</math></td></tr> <tr> <td>1</td><td><math>0.995 H_t</math></td></tr> <tr> <td>2</td><td><math>0.99 U_{total}</math></td></tr> <tr> <td>3</td><td><math>0.9999 U_{total}</math></td></tr> </table>                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | <u>mode</u> | <u>Result</u> | 0 | 1.0% change in $U_{total}$                                                                            | 1 | $0.995 H_t$                                             | 2 | $0.99 U_{total}$ | 3 | $0.9999 U_{total}$                      |   |             |   |             |   |             |
| <u>mode</u> | <u>Result</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |             |               |   |                                                                                                       |   |                                                         |   |                  |   |                                         |   |             |   |             |   |             |
| 0           | 1.0% change in $U_{total}$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |             |               |   |                                                                                                       |   |                                                         |   |                  |   |                                         |   |             |   |             |   |             |
| 1           | $0.995 H_t$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |             |               |   |                                                                                                       |   |                                                         |   |                  |   |                                         |   |             |   |             |   |             |
| 2           | $0.99 U_{total}$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |             |               |   |                                                                                                       |   |                                                         |   |                  |   |                                         |   |             |   |             |   |             |
| 3           | $0.9999 U_{total}$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |             |               |   |                                                                                                       |   |                                                         |   |                  |   |                                         |   |             |   |             |   |             |

*Continued on next page*

**Table 2: Non-Production Test Options** (*Continued*)

| number      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |             |                         |       |                                                                                                                   |       |                                                                                      |   |                                                                                          |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|-------------------------|-------|-------------------------------------------------------------------------------------------------------------------|-------|--------------------------------------------------------------------------------------|---|------------------------------------------------------------------------------------------|
| 163         | Criteria for defining $F_{max}$ in Baldwin-Lomax model ( <b>blomax</b> )<br><table> <tr> <th><u>mode</u></th><th><u>Result</u></th></tr> <tr> <td><math>&gt; 0</math></td><td>Search outward from wall; use first peak in <math>F</math> that is followed by a fractional decrease in <math>F</math> of <math>mode/1000</math>.</td></tr> <tr> <td><math>&lt; 0</math></td><td>Use the max <math>F</math> value between the wall and the <math> mode </math>'th grid point from the wall</td></tr> </table> <p>The default is +700.</p> | <u>mode</u> | <u>Result</u>           | $> 0$ | Search outward from wall; use first peak in $F$ that is followed by a fractional decrease in $F$ of $mode/1000$ . | $< 0$ | Use the max $F$ value between the wall and the $ mode $ 'th grid point from the wall |   |                                                                                          |
| <u>mode</u> | <u>Result</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |             |                         |       |                                                                                                                   |       |                                                                                      |   |                                                                                          |
| $> 0$       | Search outward from wall; use first peak in $F$ that is followed by a fractional decrease in $F$ of $mode/1000$ .                                                                                                                                                                                                                                                                                                                                                                                                                       |             |                         |       |                                                                                                                   |       |                                                                                      |   |                                                                                          |
| $< 0$       | Use the max $F$ value between the wall and the $ mode $ 'th grid point from the wall                                                                                                                                                                                                                                                                                                                                                                                                                                                    |             |                         |       |                                                                                                                   |       |                                                                                      |   |                                                                                          |
| 164         | Iteration interval for updating gas properties and species for <b>ireal</b> = 1 and 2. The default value is 1. ( <b>tdgas1</b> )                                                                                                                                                                                                                                                                                                                                                                                                        |             |                         |       |                                                                                                                   |       |                                                                                      |   |                                                                                          |
| 165         | Sets the tolerance for the distance between grid points used to determine a singular direction to $10^{mode/1000}$ . The default is a tolerance of $10^{-8}$ (i.e., $mode = -8000$ ). ( <b>bcsing</b> , <b>direct</b> )                                                                                                                                                                                                                                                                                                                 |             |                         |       |                                                                                                                   |       |                                                                                      |   |                                                                                          |
| 168         | For the algebraic turbulence models, begin turbulent flow at $i = mode$ ( <b>tdvis1</b> )                                                                                                                                                                                                                                                                                                                                                                                                                                               |             |                         |       |                                                                                                                   |       |                                                                                      |   |                                                                                          |
| 170         | NASA Ames time step formula. (CFL increases as $1/\sqrt{\Delta y}$ near the wall. Thus, $\Delta t$ decreases as $\sqrt{\Delta y}$ , not $\Delta y$ as the default.) $C_A$ is scalar coefficient on CFL; i.e., $CFL_{wall} \propto C_A$ . This test option has no effect if <a href="#">TEST 105</a> mode 1, 2, or 3 is used. ( <b>tdtmst</b> )<br><table> <tr> <th><u>mode</u></th><th><u><math>C_A</math></u></th></tr> <tr> <td>1</td><td>0.01</td></tr> <tr> <td><math>n</math></td><td><math>0.001n</math></td></tr> </table>       | <u>mode</u> | <u><math>C_A</math></u> | 1     | 0.01                                                                                                              | $n$   | $0.001n$                                                                             |   |                                                                                          |
| <u>mode</u> | <u><math>C_A</math></u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |             |                         |       |                                                                                                                   |       |                                                                                      |   |                                                                                          |
| 1           | 0.01                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |             |                         |       |                                                                                                                   |       |                                                                                      |   |                                                                                          |
| $n$         | $0.001n$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |             |                         |       |                                                                                                                   |       |                                                                                      |   |                                                                                          |
| 172         | Turn off base energy for <b>ireal</b> = 1 ( <b>aichem</b> , <b>aijkrg</b> , <b>gas1</b> , <b>gas2</b> , <b>gas3</b> , <b>gas4</b> , <b>gasint</b> , <b>pstinp</b> , <b>stomp</b> , <b>tdimfp</b> , <b>tdroe4</b> , <b>uspeci</b> )                                                                                                                                                                                                                                                                                                      |             |                         |       |                                                                                                                   |       |                                                                                      |   |                                                                                          |
| 174         | For the algebraic turbulence models, the iteration interval for updating the turbulent viscosity. The default is 1. ( <b>tdvis1</b> )                                                                                                                                                                                                                                                                                                                                                                                                   |             |                         |       |                                                                                                                   |       |                                                                                      |   |                                                                                          |
| 175         | Boundary conditions at freestream radial outer boundaries ( <b>nzn</b> = -6). ( <b>bcfree</b> )<br><table> <tr> <th><u>mode</u></th><th><u>Result</u></th></tr> <tr> <td>0</td><td>Compute characteristics from freestream conditions</td></tr> <tr> <td>1</td><td>Compute characteristics from conditions at <math>i = 1</math> along boundary</td></tr> <tr> <td>2</td><td>Extrapolate without testing at <math>k</math> boundaries; treat <math>i</math> and <math>j</math> boundaries as in mode 1</td></tr> </table>               | <u>mode</u> | <u>Result</u>           | 0     | Compute characteristics from freestream conditions                                                                | 1     | Compute characteristics from conditions at $i = 1$ along boundary                    | 2 | Extrapolate without testing at $k$ boundaries; treat $i$ and $j$ boundaries as in mode 1 |
| <u>mode</u> | <u>Result</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |             |                         |       |                                                                                                                   |       |                                                                                      |   |                                                                                          |
| 0           | Compute characteristics from freestream conditions                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |             |                         |       |                                                                                                                   |       |                                                                                      |   |                                                                                          |
| 1           | Compute characteristics from conditions at $i = 1$ along boundary                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |             |                         |       |                                                                                                                   |       |                                                                                      |   |                                                                                          |
| 2           | Extrapolate without testing at $k$ boundaries; treat $i$ and $j$ boundaries as in mode 1                                                                                                                                                                                                                                                                                                                                                                                                                                                |             |                         |       |                                                                                                                   |       |                                                                                      |   |                                                                                          |

*Continued on next page*



**Table 2: Non-Production Test Options** (*Continued*)

| number | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 176    | Sets the flow direction when the <b>ARBITRARY INFLOW</b> keyword is used<br><u>mode</u> <u>Direction</u><br>0        At the angles of attack and yaw relative to the Cartesian $x$ direction as specified in the <b>ARBITRARY INFLOW</b> keyword block<br>1        Normal to the boundary face<br>2        The $\xi$ (i.e., the $i$ index) direction<br>With modes 1 and 2, any angle of attack or yaw specified in the <b>ARBITRARY INFLOW</b> keyword block (or with the <b>FREESTREAM</b> keyword if <b>UNIFORM</b> is specified without parameters under <b>ARBITRARY INFLOW</b> ) is ignored. ( <b>bcfree</b> ) |
| 177    | Freeze maximum residual ( <b>lpgrp</b> )                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 178    | Insert freestream species buffer during <b>BLOW VALVE</b> relaxation. <sup>29</sup> ( <b>bcbled</b> )                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 179    | Modify solidbody rotation radius to get linear swirl profile in $r$ , but with zero velocity not at center of rotation. $mode = 1000r_0$ , where $r_0$ is the radius (from the point $x_c, y_c, z_c$ specified using the <b>SOLIDBODY</b> keyword in the <b>ARBITRARY INFLOW</b> keyword block) for zero velocity. ( <b>rotat</b> )                                                                                                                                                                                                                                                                                  |
| 180    | Defines the radius of the solidbody rotation core for actuator disk free vortex modeling. $mode = 1000r_{core}$ , where $r_{core}$ is the solidbody core radius. ( <b>rotat1</b> )                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 181    | ?? ( <b>evrwbd</b> , <b>wbnd2</b> )                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 182    | ?? ( <b>tdbcgs</b> )                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 185    | ?? ( <b>gasint</b> , <b>mpinit</b> )                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 187    | $mode/1000 =$ factor for suppression of streamwise pressure gradient when marching. The default is 950, and values below 800 are not recommended. When separation or strong adverse pressure gradients are causing problems, values between 800 and 900 will really help. ( <b>rhsmar</b> , <b>tdimfp</b> )                                                                                                                                                                                                                                                                                                          |
| 189    | If <b>RHS PHYSICAL FIRST</b> is specified <b>TEST 189</b> must also be specified. ( <b>prtinp</b> )                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 190    | Outgoing wave Roe boundary treatment ( <b>pstinp</b> , <b>roecof</b> , <b>roeht</b> , <b>tdbcgs</b> )<br><u>mode</u> <u>Result</u><br>0, 1      Use normal Roe boundary treatment (uses boundary point in formulation)<br>2        Lower order by one (does not use boundary point in formulation). This option cannot be used with <b>TVD</b> in the same zone.<br>3        Use zero-order extrapolation                                                                                                                                                                                                            |
| 192    | Save metrics in a temporary file. After the first cycle, metrics will be read rather than computed. This eliminates the CPU resources required to re-compute the metrics each cycle, but adds significant I/O to the computation. In the past, on at least some Cray systems, this reduced the CPU time by approximately $40.8 \mu\text{sec}/(\text{node-cycle})$ . On the more common platforms, however, it is generally faster to re-compute the metrics rather than store them. ( <b>lpcycl</b> )                                                                                                                |

*Continued on next page*<sup>29</sup>Note, however, that **BLOW VALVE** does not work in the current version of WIND.

**Table 2:** Non-Production Test Options (*Continued*)

| number | Description                                                                                                                                                                                                                |
|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 193    | Do <i>not</i> stop if a singular line is encountered normal to a wall ( <code>bbdamp</code> , <code>blinit</code> , <code>key2</code> )                                                                                    |
| 194    | Bypass singular viscous metric check ( <code>dsolv</code> , <code>vismet</code> )                                                                                                                                          |
| 196    | Overlapping grid: print an error message if there are no points adjacent to a fringe point ( <code>inttur</code> , <code>tdbcgs</code> )                                                                                   |
| 197    | Roe self-coupling mode ( <code>pstinp</code> )                                                                                                                                                                             |
|        | <u>mode</u> <u>Result</u>                                                                                                                                                                                                  |
|        | 0      Once per iteration, using <code>bcself</code>                                                                                                                                                                       |
|        | 1      Once per cycle, using standard zone coupling                                                                                                                                                                        |
| 198    | Sets the tolerance level for fuselage station <code>USERSPEC</code> inflow to <code>mode/1000</code> . The default tolerance is 0.01 (i.e., <code>mode = 10</code> ). (Currently deactivated) ( <code>uspeci</code> )      |
| 199    | Singular axis averaging — average from 1 to ( <code>max - 1</code> ), <i>not</i> 1 to <code>max</code> . ( <code>bcsing</code> , <code>linzero</code> , <code>radavg</code> , <code>relsng</code> , <code>sngthrm</code> ) |
| 200    | Don't bomb for negative speed of sound in <code>tdroe3</code> ( <code>tdroe3</code> )                                                                                                                                      |

## References

- Baldwin, B. S., and Lomax, H. (1978) "Thin Layer Approximations and Algebraic Model for Separated Turbulent Flows," AIAA Paper 78-257.
- Baldwin, B. S., and Barth, T. J. (1990) "A One-Equation Turbulence Transport Model for High Reynolds Number Wall-Bounded Flows," NASA TM 192847.
- Bush, R. H. (1988) "A Three Dimensional Zonal Navier-Stokes Code for Subsonic Through Hypersonic Propulsion Flowfields," AIAA Paper 88-2830.
- Chien, K. Y. (1982) "Prediction of Channel and Boundary-Layer Flows with a Low-Reynolds-Number Turbulence Model," AIAA Journal, Vol. 20, No. 1, pp. 33–38.
- Chung, J., and Cole, G. L. (1996) "Comparison of Compressor Face Boundary Conditions for Unsteady CFD Simulations of Supersonic Inlets," NASA TM 107194.
- Cornell, W. G. (1958) "Losses in Flow Normal to Plane Screens," Transactions of the ASME, May 1958, pp. 791–799.
- Mani, M., Ladd, J. A., Cain, A. B., and Bush, R. H. (1997) "An Assessment of One- and Two-Equation Turbulence Models for Internal and External Flows," AIAA Paper 97-2010.
- Mayer, D. W., and Paynter, G. C. (1994) "Boundary Conditions for Unsteady Supersonic Inlet Analyses," AIAA Journal, Vol. 32, No. 6, pp. 1200–1206.
- McLafferty, G., and Ranard, E. (1958) "Pressure Losses and Flow Coefficients of Slanted Perforations Discharging from Within a Simulated Supersonic Inlet," United Aircraft Corporation, Report R-0920-1, Dec. 1958.
- Menter, F. R. (1993) "Zonal Two Equation  $k$ - $\omega$  Turbulence Models for Aerodynamic Flows," AIAA Paper 93-2906.
- Nichols, R. H., and Tramel, R. W. (1997) "Application of a Highly Efficient Numerical Method for Overset-Mesh Moving Body Problems," AIAA Paper 97-2255.
- Paynter, G. C. (1998) "Modeling the Response from a Cascade to an Upstream Convective Velocity Disturbance," AIAA Paper 98-3570.
- Rodi, W., and Scheuerer, G. (1986) "Scrutinizing the  $k$ - $\epsilon$  Turbulence Model Under Adverse Pressure Gradient Conditions," Transactions of the ASME Journal of Fluids Engineering, Vol. 108, pp. 174–179.
- Romer, W. W., and Bush, R. H. (1993) "Boundary Condition Procedures for CFD Analyses of Propulsion Systems — The Multi-Zone Problem," AIAA Paper 93-1971.
- Sajben, M. (1999) "Prediction of Acoustic, Vorticity and Entropy Waves Generated by Short-Duration Acoustic Pulses Incident on a Blade Row," ASME Paper 1999-GT-148.
- Shur, M., Spalart, P. R., Strelets, M., and Travin, A. (1999) "Detached-Eddy Simulation of an Airfoil at High Angle of Attack," Fourth International Symposium on Engineering Turbulence Modeling and Measurements, May 24–26, 1999, Corsica.
- Slater, J. W., and Paynter, G. C. (2000) "Implementation of a Compressor Face Boundary Condition Based on Small Disturbances," ASME Paper 2000-GT-0005.

Spalart, P. R., and Allmaras, S. R. (1992) "A One-Equation Turbulence Model for Aerodynamic Flows," AIAA Paper 92-0439.

Spalart, P. R., Jou, W. H., Strelets, M., and Allmaras, S. R. (1997) "Comments on the Feasibility of LES for Wings, And on a Hybrid RANS/LES Approach," First AFOSR International Conference On DNS/LES, Aug. 4–8, 1997, Ruston, Louisiana. In *Advances in DNS/LES*, Liu, C., and Liu, Z., eds., Greyden Press, Columbus, Ohio.

Syberg, J., and Hickox, T. E. (1972) "Design of a Bleed System for a Mach 3.5 Inlet," NASA CR-2187, Sept. 1972.

Thomas, P. D. (1979) "Numerical Method for Predicting Flow Characteristics and Performance of Nonaxisymmetric Nozzles — Theory," NASA CR 3147.

Tramel, R. W., and Nichols, R. H. (1997) "A Highly-Efficient Numerical Method for Overset-Mesh Moving-Body Problems," AIAA Paper 97-2040.